DOCUMENT RESUME

ED 263 902 IR 011 889

TITLE	Computing Knows No Borders. Proceedings of the Joint Conference of the Association for Educational Data Systems and the Educational Computing Organization of Ontario (Toronto, Ontario, Canada, April 21-26, 1985).
INSTITUTION	Association for Educational Data Systems, Washington, D.C.; Educational Computing Organization of Ontario, Toronto.
REPORT NO	ISBN-0-9692016-0-5
PUB DATE	85
NOTE	357p.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - General (140)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Computer Assisted Instruction; *Computer Literacy; *Computer Software; Conference Proceedings; Elementary Secondary Education; *Microcomputers; Programing; *Programing Languages; *Special Education; Technological Advancement
IDENTIFIERS	*Computer Uses in Education; Logo Programing Language

ABSTRACT

The 88 papers included in this publication are organized into seven groupings used as themes in the conference, "Computing Knows No Borders." The topics are: Administration (12 papers); Computer Studies (13 papers); Educational Tools (10 papers); Logo (7 papers); Software (9 papers); Special Education (8 papers); and Teaching (16 papers). An eighth group under the heading "Other" includes 13 papers which cut across several of the other themes and therefore deserved more general recognition. An author index is provided. (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ASSOCIATION FOR EDUCATIONAL DATA SYSTEMS
EDUCATIONAL COMPUTING ORGANIZATION OF ONTARIO

ECOO/AEDS CONFERENCE PROCEEDINGS

"COMPUTING KNOWS NO BORDERS"

APRIL 21-26, 1985

HARBOUR CASTLE HILTON HOTEL
AND
ONTARIO INSTITUTE FOR STUDIES IN EDUCATION
TORONTO, ONTARIO

ED263902

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Yvonne M. Armstead

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR011889

Association for Educational Data Systems
1201 Sixteenth Street N.W.
Washington, D.C. 20036

Educational Computing Organization of Ontario
252 Bloor Street West
Toronto, Ontario M5S 1V5

Copyright 1985, Educational Computing Organization of Ontario and
the Association for Educational Data Systems
ISBN 0-9692016-0-5

The cover --

The cover picture is a view of the city of Toronto taken from the west. The city sits on the shores of Lake Ontario -- the foreground shows the geodesic dome of Ontario Place and some of the islands which create an enclosed harbour. The skyscape of Toronto is dominated by the distinctive shape of the CN Tower - the highest free-standing structure in the world. The many tall buildings of the business area stand behind the residential and recreational areas which front on the lake. This city of 2 million people, and the capital of Ontario, is a favourite convention centre and a shopping and holiday choice.

3

TABLE OF CONTENTS

Foreward v
Table of Contents iii
1984 Conference Planning Committee vii
ECOO Board of Directors viii
AEDS Board of Directors ix
Index of Papers xiii
Papers xxi
List of Papers by Author 343

FOREWARD

The 1985 ECOO/AEDS Conference, "Computing Knows No Borders" is a unique joint effort between two educational computing organizations. AEDS (The Association for Educational Data Systems) has offered twenty-seven previous conventions for its members in a variety of locations across the United States - each annual session coordinated by the local AEDS chapter. ECOO (The Educational Computing Organization of Ontario) is the largest organization in Canada serving computing educators and annually runs a major conference in the Toronto area.

The conference planning committee, consisting of both MAEDS (Michigan AEDS) and ECOO representatives, attempted to prepare a package that would appeal to educators using computers at all levels and with all measures of experience. The resulting programme offered 48 workshops including hands-on activities, tours of industrial locations and mini-courses in topics of particular relevance. At the two conference sites, the Harbour Castle Hilton Convention Centre and the Ontario Institute for Studies in Education, over 400 presentations were spread through the four session days - over 25 choices in every time slot.

These Proceedings represent only a portion of the material prepared for presentation at the conference. Many of the sessions were difficult to put together in a meaningful paper because of the hands-on or practical nature of the topic or the multi-media format of the message. The 88 papers included in this publication are organized in the seven groupings used as themes in the conference structure - Administration, Computer Studies, Educational Tools, Logo, Software, Special Education, Teaching, and Other. The last group includes papers which cut across several of the other themes and therefore deserved a more general recognition. For ease of use, the Proceedings also have a list by author at the end of the papers.

As with most proceedings publications, the work submitted by the authors was printed as it arrived with the exception of the commonly formatted titles. Each author(s) is responsible for the quality of the presentation and content of the paper bearing their name. For many members of both ECOO and AEDS, preparing papers for a conference proceedings is a new experience. It is not a difficult task, but should be done with care and precision if the document is to be one of which the authors and the two organizations will be proud. To help members who may in the future plan on submitting papers for publication, the set of instructions sent to all this year's presenters has been included as the last paper in this proceedings.

Because of the uniqueness of the conference, this Proceedings is catalogued with an ISBN separate from the series ISSN or Library of Congress numbers for previous Proceedings. Both organizations will return to the traditional ISSN designations in subsequent years.

While the difficulties of bringing together the efforts of two major and distinct groups such as AEDS and ECOO made the conference planning a special kind of adventure, the conference theme "Computing Knows No Borders" was borne out as the sameness of the issues, the solutions and the characteristics of the people involved became increasing apparent. Certainly effort, commitment and quality in the support of computer use in education knows no borders!

Jo Ann Wilton
Conference Co-Chairperson and Proceedings Editor

COMPUTING KNOWS NO BORDERS
1984 CONFERENCE PLANNING COMMITTEE

Conference Co-Chair:	Jo Ann Wilton, Ontario Institute for Studies in Education, Toronto Tom Schmeltzer, IBM Corporation, Georgia (co-chair until October 1, 1984) Lary Smith, Wayne County ISD, Wayne MI (co-chair after Dec. 20, 1984)
Assistant Co-Chair:	Trudy Van Buskirk, SES (Computing) Inc., Toronto
Program Committee:	Dave Didur, Ontario Ministry of Education: co-chair Dick Alverson, Dearborn Resource Center: co-chair Assistants: Nancy Murray Terrence Coburn Carol Klenow Al Monetta Susan Moont Noelle Lucas Lynn Liebermann Jan Van Dam
Social Committee:	Colin Harvey, Upper Canada College: co-chair Harry Groves, Windsor Board of Education: co-chair
Finance Committee:	Guy Leger, Metropolitan Separate School Board: co-chair Lary Smith, Wayne County ISD: co-chair Jim Rarus, Wayne County ISD
Publicity Committee:	Alan Winarski, Halton Board of Education: co-chair Carol Klenow, Oakland Schools: co-chair
Exhibits:	Brock Godfrey, Ontario Ministry of Education: chair
Registration:	Bob McLean, Ontario Institute for Studies in Education: chair
Equipment:	Art Jewell, Toronto Board of Education
Evaluation:	Ron Ragsdale, Ontario Institute for Studies in Education
Liaison:	Irvin Rubincam, ECOO/AEDS liaison Dave Brittain, AEDS convention liaison
Secretary:	Gordon McKye, Etobicoke Board of Education
Hilton Site Manager:	Steven Lee, Peel Board of Education
Conference Manager:	Irene Leps, OISE Conference Office

ECOO BOARD OF DIRECTORS
1984-1985

President Mr. Alan Winarski
Halton Board of Education
416/878-2743

Past President Ms. Jo Ann Wilton
Peel Board of Education
279-6010 x334

President Elect Mr. Gordon McKye
Etobicoke Board of Education
416/626-4360

First Vice President Ms. Trudy Van Buskirk
SES (Computing) Inc.
416/366-6761

Second Vice President Ms. Susan Moont
Ontario Institute for Studies in Education
416/365-0838

Secretary Mr. Paul Rehak
Hamilton Board of Education
416/547-2574

Treasurer Ms. Muriel Fung
OISE/MECA
416/923-6641

1984-1985 Directors:

Ms. Karen Bullock
Carleton Board of Education
613/820-1820

Dr. Dale Burnett
Faculty of Education, Queen's University
613/547-6971

Dr. Don Cowan
University of Waterloo
519/885-1211 x3238

Mr. Terry Cowx
London Board of Education
519/471-6396

Mr. Dave Didur
Ontario Ministry of Education
416/965-0692

Mr. Des Dixon
Ontario Teachers' Federation
416/966-3424

Mr. Brock Godfrey
Ontario Ministry of Education
705/648-4326

Mr. Colin Harvey
Upper Canada College
416/488-1129

Mr. Gerald Hurtubise
Ontario Ministry of Education
613/225-9210

Mr. Bob Jones
Sheridan College
416/845-9430 x204

Mr. Guy Leger
Metro Separate Board of Education
416/222-8282 x2443

Ms. Carol Lyons
Fort Frances-Rainy River District RCSSB
807/274-7747

Dr. Nancy Murray
Windsor Separate School Board
519/253-2481

Mr. George McRae
Halton Board of Education
416/336-3663

Mr. Douglas McDougall
Waterloo Board of Education
519/885-4620

Mr. Stan Squires
Harbourfront Computer Centre
416/366-2045

Dr. Ron Ragsdale
Ontario Institute for Studies in Education
416/923-6641

AEDS BOARD OF DIRECTORS
1984-1985

President Mr. Stephen M. Raucher
Montgomery Co. Public Schools
301/279-3581

President-Elect Ms. Carolyn Meinhardt
WORDWARE
612/543-2294

Treasurer Dr. Johnette Hassell (1986)
Tulane University
504/865-5480

Past President Mr. John W. Kelly
Palm Beach Junior College
305/439-8040

Secretary Ms. C. Dianne Martin (1987)
George Washington University
202/676-8238

1984-1985 Directors:

Ms. Carol Edwards (ex-officio)
So. Coalition for Educational Equity Inc.
404/522-5566

Dr. John Hamben (ex-officio)
University of MO - Rolla
304/341-4836

Dr. Peggy Roblyer (1987)
Florida A&M University
904/599-3022

Dr. Charles Speiker (ex-officio)
Millard Public Schools
402/895-8214

Dr. John Stiglmeier (1987)
New York State Educ. Dept.
518/474-8716

Ms. Ann Tompkins (1985)
San Diego Public Schools
619/293-8379

Mr. Donald Friermuth (ex-officio)
Tacoma School District
206/593-6915

Mr. Alan Olkes (1985)
Dade County Public Schools
305/559-7050

Mr. Lary Smith (1985)
Wayne County ISD
313/467-1558

Dr. Dennis Spuck
University of Houston
713/749-7281

Ms. Pat Sturdivant (1985)
Houston ISD
713/960-8888

Executive Director (Staff): Shirley Hammond, 202/822-7845

MAEDS Board of Directors
1984-1985

President Mr. Harry Groves
Past President Mr. Tim Palazzolo
Secretary Ms. Nancy Wellman

President-Elect Mr. Stan Bittner
Treasurer Ms. Joan Rathbun

Board Members:

Dave Rhodes
Mary Louise Kennedy
Dave Classon

Nicholas W. Colovos
Lary R. Smith
Terry Henderson

AEDS Affiliate Presidents and Representatives

Functional

Information Processing Administrators
of Large School Systems
Donald Friermuth
Tacoma School District
206/593-6915

Geographic Affiliate Rep

Charles Speiker
Millard Public Schools
402/895-8216

Geographic

Alaska
Sue Henry
Dept of Education
907/456-4685

Alberta
Ralph Krueger
University of Calgary
904/284-6493

California Ed'l DP Association
John Kern
San Francisco USD
415/565-9247

Greater Delaware AEDS
Roland Garton
U of Delaware
302/451-8161

District of Columbia
Gordon Lewis
Institutional Services Ctr
202/575-7816

Florida
Bernie Slessinger
U. of Central Florida
305/275-2000

Illinois
Ellie Gillespie
DePaul University
312/321-7675

Iowa
John Cook
AEA 6 Ctr.
515/752-1578

Manitoba
Cheryl Prokopanko
Manitoba Dept of Ed.
204/945-6793

Maryland
John Randall
Montgomery P.Schools
301/279-3321

Michigan
Harry Groves
Windsor Bd of Education
519/255-3300

Mid-America
John Wolverton
Wichita Public Schools
316/263-0325

Mid-South
Mike Schouest
LA Dept of Education
504/342-1809

Minnesota
Earl Keyser
TIES
612/638-2346

Nebraska
Charles Friesen
Lincoln Public Schools
402/475-1081

New Jersey
Harry Selover
Newton Public Schools
201/383-1900 x502

New York
Eileen Gress
Fairport Central School
716/223-5947

North Carolina
Gene Bailey
E. Tennessee State U.
615/929-5332

Ohio
Larry Wilberding
Shelby County Schools
513/498-2161

Oregon
Mike Stevenson
Mt. Hood Comm Coll
503/667-7298

Rocky Mt.
Pat Leonard
Jefferson Public Schools
303/987-5500

Tennessee
Dr. Gary Ubben
U. of Tennessee
615/974-2214

Texas
Dr. Herbert Rebhun
U. of Houston-Downtown
713/221-8017

Virginia
H. Gilpin Brown
U. of Richmond
804/285-6494

Wisconsin
Roland Hicks
CESA 6
715/723-0341

INDEX OF PAPERS

ADMINISTRATION

Retooling Black Schools of Business for the Information Culture 1
Evans, C.L.

Expert Systems:Applications in Educational Research and Decision Making 5
Bozeman, Wm. C. and Olsen, C.R.

Where Would You Go Tomorrow if Disaster Struck Today? 9
Hadduck, C.G.

From Course Requests to Examination Schedule 13
Osborne, J.

Administrator "Homo-Optimus": A Diagnosis of Change and Relationships Affected by Technology 17
Goldstein, G.

A System and Strategy for Data Processing Planning for Universities 21
Dawley, D.L.

Managing Information: Security, Access and Dissemination - A Conceptual Model and Case Study 25
Bhagaloo, R.K.

Linking Microcomputers to Mainframes for Period Attendance and Local School Flexibility 29
Goates, W.

A System for Computer-Managed Instruction 32
Hadduck, C.G.

Getting to the Core of the Apple: Global Computer Development 35
Kreutz. D.A.

Planning Computer Labs 38
Edwards, L.

Computerizing the Classroom Teacher's Tasks (Burn Your Computer Out - Not Yourself!) 42
Crawford, R.M.

COMPUTER STUDIES

Graduate Microcomputer Education: Another Perspective 47
Anderson, E.L.

What are the Effects of Computer Programming on Cognitive Skills? 51
Land, M. and Turner, S.

QNX and the ICON - Security that Works 55
Trueman, D.

Training Teachers of Secondary Computer Science - A Methods Course 59
Kepner, H.S.

Computer Studies in Ontario Schools: Curriculum, Hardware and Software 63
Didur, D.

Kids and Computers - A Summer Program for Elementary Students 67
Cieply, S.M.

Identifying Student Characteristics Influencing Success in Introductory Computer Science Courses 69
Oman, P.W.

Programming Schemes: A Model for Understanding and Teaching Programming, Programming Debugging and Testing 73
Hassell, J.

An Evaluation of the Course Content of a CIS Curriculum. 77
Hunter, J.

Using Low-Resolution Graphics to Teach B.A.S.I.C. 81
Lapp, S.

The ALICE Programming Education System 85
Templeton, B. and Gardner, J.

Elementary Computer Literacy: An Inexpensive Strategy. 89
Good, J.

Acquiring Computer Literacy in Higher Education: A Tennessee Mandate 93
Bailey, M.G.

EDUCATIONAL TOOLS

Authoring Language/Systems in K-12 Schools: What's Happening 101
Burger, M.L.

Second Generation Electronic Spreadsheets and the Teacher's Gradebook 105
Spero, S.W.

Computers and Composition in the Music Education Classroom 109
Hermann, B.

Talking Books - Print Learning Materials Go Interactive 113
Nitefor, M.

Integrating Tool Software in the School and Classroom 117
Burton, J.

Computing Tools in the Classroom: Through the Curriculum and Beyond 121
Olds, H.F.

Competency Based Computer Education 124
Sullivan, G.

Spreadsheets and Decision Support Systems: Tools and Environments for Teaching and Learning 127
Marchionini, G.

Oracle (Consultant) Science Fair Research Project 130
Byrnes, R.

Databases, Bulletin Boards and Information Retrieval: Promises and Pitfalls (Or Should I Buy a Modem?) 133
Moont, S.E.

LOGO

Teacher Training and Logo: Higher Levels of Concern 139
Martin, C.D. and Heller R.S.

Creating Logo Microworlds 143
Watt, D.H.

It's Almost Time for Logo 147
Burnett, J. D.

Teaching Recursion in Logo by Playing Turtle and Playing Procedure 151
Watt, M.

The OISE Logo Project: Apple/Terrapin Software 154
Cohen, R. and Barker, G.

Developmental Studies of Children Learning Logo Programming:
The Cognitive Dimension 157
Cohen, R.

Robotics with Logo and a Tasman Turtle 161
Goforth, D.

SOFTWARE

Alcohol Education: The Choice is Yours 167
Thomas, B.S.

Use of CAI to Improve Instruction in Nursing Research 171
Thomas, B.S.

Marketing Educational Software: A Publisher's Perspective. 175
Paine, J.

Electronic Thinking Cap: Microcomputer Enhanced Creative Problem Solving
for Visually Oriented Students 179
Metros, S.E.

Melodic Manipulations: Teaching Composition with Computers 183
Upitis, R.

It May Be Courseware, But Is It Educational? 187
Durell, B.

Custom-designed Lessonware for Second Language Learning 191
Paramskas, D.M.

Micrometrics: A Metric System Tutorial 194
Malone, A.P.

A Model for Implementing Computers (K-12) to Promote Learning and Literacy 197
Rivers, R. and Vockell, E.

SPECIAL EDUCATION

Individual Educational Plans - A Continuing Saga of Computer Support 201
Koehler, T.J., Barr, D., Edmister, P. and Bowers, A.

Computers and Special Students: Classroom Applications Beyond Drill and Practice 205
Bederman, F.

Word Processors or Story Processors? (Redefining Literacy) 209
Turnbull, G.R.

Using Microcomputers to Improve Self Image of Learning Disabled Students. 213
Speed, B.

On-line Needs Diagnosis. 217
McCahill, P.

Some Thoughts on Data Management for Special Education. 220
Benstein, J.H.

An Information Age Plan for Special Educators: Computers and Technology 223
Tabor. F.M.

Introducing Computers into a T.M.R. School: a Proposal. 226
Robinson, S.

TEACHING

Looking at Learning from Lessonware 231
Pike, R.

Problem Solving: A Computer Oriented View. 235
Moursund, D.

It Figures: The Computing Component of the Second International Mathematics Study 239
Goddard, Wm.P.

A Model for Inservice Training of Teachers That Effects Real Change 243
Burns, J.

An Assessment of CAI in the Field of ESL 247
Freeman, R.J.

Incorporating Computer Simulation Methods Into Classroom Instruction: From Social Studies and Mathematics to Emergency Medicine 251
Wholeben, B.E.

Local Area Networks: Present and Future Possibilities for Classroom Use 255
Wilton, J.A.

Human Factors Concerns in Educational Software Design, Selection and Evaluation 259
Morariu, J.

An Analysis of Instruction Improvement Projects Utilizing Computer Resources 263
Gillespie, F.

Diary of a Computer Resource Teacher - Year One 266
Matthews, J.E.

On-line Data Bases: A School Project 269
Levinson, M.S. and Walcott, J.A.

Computers and Art . . 272
Noonan, L.

Computer Applications Across the Business Education Curriculum 275
Anrep-Matthiesen, F-G.

Strategies for Developing Faculty Computer Literacy 277
Frye, I. and Parsons, J.A.

Minnesota Information Technology Learner Outcomes 279
MacDermot, Dr. H.G.

"MacEnglish" - Using the Macintosh for Senior High English and Journalism 283
King, T. and Parish, M.

?! OTHER

"Clustering" in Vocational Business Education 289
Avani,N.

Tips for Teachers on Organizing Data 293
Fung,M.

Research Needs for Computers in Education 297
Ragsdale, R.

Constructing Computer Interfaces for the Laboratory 301
Rouse, B.

Computer Attitudes and Knowledge Among Students and Educators 305
Marshall, J.C. and Bannon, S.H.

Documentation - - A Key to Success 309
Meinhardt, C.

Educational Computing in Britain 313
Brine, J.W. and Johnson, E.M.

Characteristics of the Periodical Literature in Computer Applications to Education 318
Rubincam, I. and Somer. G.

Sexism, Computers and Young Children 323
Woodill, G., Anderson, K. and Bernhard, J.

Must Machines Mechanize?: Human Issues in Educational Computing. 326
Higginson, Wm.

Educational Programming in 'c': A Toolkit Approach 330
McLean, R.S.

Review of "Learning and Telecommunications in New York State: Action Recommendations" 335
Stoll, P.

Instructions on How to Prepare Papers for Conference Proceedings 339
Wilton, J.A.

ADMINISTRATION

Retooling Black School Of Business For The Information Culture

Charles L. Evans
Florida A&M University
School of Business and Industry
Tallahassee, Fl. 32307

There is a need for Black schools of business to be retooled. This can be accomplished by training its faculty, acquiring appropiate hardware and software, and changing its curriculum. Both the school and students would receive educational and financial benefits from this retooling.

Introduction

A massive retooling of the managerial skills and techniques employed by managers in American companies is underway, prompted by fundamental changes in computer economics and the continuing advancement of the technology itself. An integral aspect in the development of students in Black schools of business entails improving their capacity to relate to changing business practices and managerial needs in the information revolution. The information revolution is spreading throughout the business world and Black business schools are feeling the crunch. Electronic technology has dramatically revised the way all forms of business operate. Consequently, Black schools must effectively integrate the computer and other forms of information related technology into their curricula activities. The focus should be on the micro computer. Business managers are turning to these smaller computers for their immediate information needs.

If Black students are going to successfully function in American businesses in the future, there has to be some extensive exposure to computers. There are no easy answers for ensuring that Black students receive the necessary exposure.

While the emphasis is on retooling Black school of business, these problems and planned solutions are not unique and are common in our American education system.

Impact of Activities on Users

Throughout the educational community, students are being taught how to effectively and efficiently manage the information technology and the new technocrats being produced. These students will reshape the traditional role of the computer in American companies. Black schools must be careful to determine their needs in terms of the audiences that will be served. The needs are being influenced by three revolutions. (Ward, 1984)

First is the hardware revolution, in which the micro computers have become economically appealing. There is a shift from the central large mainframe computers to the distributed processing systems using micros.

The second revolution is software. Languages and programs are becoming more sophisticated. For micro computers, the introduction of spreadsheet, word-processing, data base management, and graphic software has catapulted it into many firms.

The third revolution is in management. A shift from the more common centralized form of management to decentralization of computing resourses is occuring.

The needs of the Black schools can generally be categorized into the following areas:

1. Instruction and curriculum related
    A. Mass use
        1) General classroom use
        2) Simulation
    B. Specialized use
        1) Computer Aided Instruction
        2) Access to external data base
        3) Wordprocessing
        4) Business graphics
2. Administration
3. Research

Planned Solutions

In education, computers have changed the very nature of many of our educational needs and have brought into question many of the basic assumptions upon which our educational process is built. In the past decade, expenditures for academic computing in higher education (research, instruction, and administration) have more than doubled. Over half of the current faculty members in our educational institutions completed their formal training before computers in large numbers appeared on the scene. (Molnar, 1981)

The question is, are the Black schools of business prepared for this revolutionary change? Can we train a sufficient number of sophisticated business computer specialists and professionals to meet a share of our national needs? Do the Black schools have the resources to train this new generation of computer literate professionals? The answers to these questions should emphasize: (1) teacher training, (2) microcomputer hardware and software for edcuation, (3) curriculum revision and development, and (4) student motivation.

(1) Professors as a change agent will be the approach used to change the education of students. A well organized training program is necessary to teach professors how to use computers effectively. Brief workshops are entirely inadequate for producing adequate teachers who understand educational uses of computers.

Teachers are an essential part of any change process in education. But teacher education by itself has limited potential. (Moursund, 1984)

(2) The existance of computer hardware and software in the schools will not guarantee the needed change to meet the demands of the electronic revolution. The time-phased installation of prepackaged application software and commercially available hardware items on an online interactive distributed processing basis promises significant changes in the curriculum and instructions. These items consist of commercially available micro-computer software, communication hardware and software, televideo conference systems, central site hardware and stand-alone computer equipment.

(3) Computers are forcing a reexamination of the curriculum in business schools. As a consequence of the reexamination, many Black schools are requiring students to take more rigurous courses in English, science, and mathematics. New courses in Management Information Systems (MIS) are being developed and offered such as Data Base Management, Problem Solving and Logic, Ethics and Privacy, and Information Systems and Management.

(4) University systems are an environment designed to facilitate learning. But what the student learns and how well he learns it is ultimately up to the student. The Black schools of business make opportunities available - it is not a panacea. Ultimately the individual student is the key. The effort to motivate students becomes a problem, we must be aware that computers have strong motivational powers for many students.

Computers can play a helpful role in an educational system of self-reliant independent learners. Over the next 20 years computers will significantly supplement books as a source of information. Computers will supplement teachers as a source of instruction, testing, and feedback. Computers will become individual tools, as pencils and paper are today, to aid in the learning and problem solving process. (Moursund, 1984)

Measurable Benefits

Not to implement programs of this nature could lead to the eventual downfall of many Black schools since competitor schools are proceeding with their type programs. Black business schools will realize the key benefits outlined below:

..Improved efficiency of the basic education process.

..Continued growth in the enrollment of higher quality students.

..Greater recruitment and retention of quality faculty.

..Reduced total school operating costs measured on a per end-user basis.

..Increased capability to expand course offerings without the unfavorable impact of linear increases in faculty and staff.

..Better and more competitive end product (graduated students) based on improved teaching methods that are technologically current.

REFERENCES

Bork, Alfred, Learning with Computers,

Digital Press, 1981

Keller, Arnold, "DP Professionals: Redefining the Role", Infosystems, October, 1983.

Forest, Robert, "Neglected Resources in the DP Industry", Infosystems, October, 1983.

Mangum, William, "Minorities Succeed in DP", Data Management, November, 1981.

Molnar, Andrew, "The Coming of Computer Literacy: Are We Prepared for it", Educational Technology, January, 1981.

Moursund, David, "To Improve Education", Creative Computing, November, 1984.

Noel, Pamela, "The Computer Revolution: Will Blacks be Left Behind?" Ebony, May, 1984.

Nolan, Richard, "Steering DP Through A Recession", The Harvard Business Review, October, 1982.

Ward, Bernie, "Executives and the Computer", Sky, December, 1984.

Expert Systems: Applications in Educational Research and Decision Making

William C. Bozeman
The University of Iowa
210 Lindquist Center
Iowa City, IA 52242

Christopher R. Olsen
Cedar Rapids School District
2205 Forest Drive, SE
Cedar Rapids, IA 52403

This paper is concerned with two principal topics: (a) the concept of expert systems and (b) the application of an expert system in education. Results of a study are reported which involved the design and field test of a microcomputer-based system designed for the purpose of facilitating the selection of appropriate statistical procedures required for certain educational problems.

Artificial intelligence has been a topic of interest to computer professionals for over three decades. Now, with the advent of very powerful, low-cost computers, applications of artificial intelligence are being seriously explored in a number of fields. This paper will introduce an application of artificial intelligence, the expert system, and discuss an experiment related to the use of expert systems in education.

Before proceeding, let us pose an old question. Is it possible for a machine to "think" and thereby provide "expert" advice? This is a question that has been the subject of much discussion and debate. In fact, one of the early objectors to the concept of a "thinking machine" was Augusta Ada Byron, Countess of Lovelace. Lady Lovelace's principal concern was that a machine is unable to do anything really new or original. Machines only do what people have instructed them to do.

The notion of a "thinking machine" was raised again in 1950 by the British mathematician, A. M. Turing, in his paper entitled "Can A Machine Think?" (Turing, 1950). Turing described a game called the "imitation game" to be played by three persons: a man, a woman, and an interrogator. The interrogator remained in a separate room apart from the man and woman and would ask questions. The object of the game was for the interrogator to determine which of the other two was the man and which was the woman. Given such a game, Turing then posed the question, "What would happen when a machine takes the part of the man in the game?" Would the interrogator make the correct decision as often when a machine is a player? Turing's question and test continues to pose a challenge.

Such questions related to artificial intelligence are no longer the province of the mathematician, philosopher, or logician. In fact, artificial intelligence (AI) is often considered to be the 5th generation of computer development.

Initial efforts in AI were related to applications such as natural-language translation and pattern translation. The efforts were largely unsuccessful and caused AI to fall into disfavor during the 1960s. Now a renewed interest is being shown in AI. This interest is due, to a great extent, to increased machine capabilities coupled with a growing concern for the improvement of human-machine interfaces. These "practical" interests in use and applications have largely replaced the earlier debates related to the ability of a machine to think.

One application of AI research is the "expert system." An expert system is a computer system which emulates the human expert, using knowledge engineering and heuristics. Stated another way, expert systems employ a knowledge base, rules or decision techniques employed by an expert in a given field, and a system of logical operations. Expert systems, therefore, are applicable in real-world situations. Expert systems may employ huge data bases and a collection of the best rules of thumb that can be applied to that data to make it yield solutions to problems. Expert systems capture, codify, and make available to others specialized human expertise and magnify it through the memory capacity and power of the computer (Hirsch, 1984).

Expert systems are being used in many areas including geology, medicine, business, science, computing, and education. PROSPECTOR evaluates sites for potential mineral deposits. MYCIN diagnoses bacterial infections and prescribes treatment. DENTRAL analyzes mass spectograms. In education, GUIDON is a CAI program that teaches the student user by eliciting and correcting answers to a series of technical questions (Webster & Miner, 1982). While none of these systems approach a model of human intelligence, they can enhance human problem-solving and decision-making capabilities.

An experiment was recently performed at The University of Iowa to assess the potential of an expert system in an educational situation. The particular type of decision situation which was emulated in the experiment was the act of choosing an appropriate hypothesis and statistical procedure.

This particular decision process was selected in the investigation of expert systems primarily because it is a significant problem in the real world. Many educators do not have an extensive mathematical/statistical background. What little training they may have had in statistics in their graduate work usually atrophies through disuse. The nature of their jobs is such that statistical procedures are rarely used. Should a problem arise which may be "solved" using statistics or statistical hypothesis testing, the educator may not have the time, or still have the necessary skill to look up the appropriate procedure or formulae. Even though there are many microcomputer-based statistical packages available, the educator must still decide which statistical technique is appropriate for the given situation. It is at this point that an expert system could be utilized. Moreover, the system must function in the face of a dearth of knowledge on the part of the user.

It was also recognized that the practical human limitations in typical educational settings are particularly salient. There is very little time for training, so the system must not be complex to use. The typical educator simply will not allocate the time to learn a set of complicated instructions. The expert system must, with ease and on a very short time line, solicit needed information from the user and make a recommendation in an easily understandable format. In the popular jargon, the expert system must be "user friendly."

In order to evaluate the feasibility of an expert system in an educational setting, a microcomputer was programmed to elicit from the user, responses to questions which govern the choice of a particular statistical hypothesis for a specific situation. One of the questions, for example, was:

> Miss Anna H. Olsen is a 4th grade teacher at P.S. #37. Some state money has been allocated for teaching gifted and talented youngsters but, in order to qualify, a class must have a mean IQ greater than 125. Miss Olsen administers the Stanford Binet IQ test to a sample of 20 of her students. The mean of the sample is 130 and the standard deviation is 8. To see whether her class qualifies for some of the state money,

Miss Olsen should use which null hypothesis?

The experimental subjects were given a "test" of similar situations and asked to match the statistical hypothesis needed from a list provided by the experimenter. After reading a problem situation the system queried the user to match the statistical hypothesis needed from a list provided by the experimenter. After reading a problem situation the system queried the user to elicit information such as number of groups, size of groups, and particular facts of interest (e.g., proportion or mean). The system, upon receipt of the necessary information, would respond with a short answer recommendation, e.g., "The appropriate statistical hypothesis in this case is that of equal means; the t statistic should be used." The experimental subjects' scores were compared with the scores of a control group. Members of the control group attempted to identify the correct statistical hypothesis without the aid of the computerized expert system. The questions of primary interest in this research were (a) the ease of use as perceived by the subjects, and (b) the effectiveness of the system, as measured by the number of correct recommendations by the system.

The results of the experiment may be easily summarized. On the question of ease of use, even people who had virtually never used a computer found it easy to use the expert system after less than five minutes of instruction. The subjects did not seem nervous or anxious at the computer, and many of them spontaneously expressed the view that use was "easy," and that they wished they had use of such a system when they were taking their statistics courses. The effectiveness of the system was dramatic. The differences between the expert system group and a control group were not only statistically significant, $t(26) = 3.83$, $p < .001$, but a large effect size was noted. This large effect occurred in a pool of users whose feelings of competence about statistics was very low; probably their self perceptions were accurate. The means of the experimental group and control group were respectively, 11.40 and 7.23 based on 18 problem situations (Olsen, 1984).

Based on this research, some tentative conclusions and recommendations for the use of expert systems in education may be noted. The first conclusion is that expert systems are certainly plausible. Subjects who were computer-naive were able to operate the system with a minimum of training, and without frustration. This is very important when one considers the audience of users in an educational setting. Such people cannot be regarded as computer literate, and have a low tolerance for machine error. People in education are typically person-, not machine-oriented; an expert system may be uncannily reliable in its advice, but if it is not extremely easy to learn and use, that accuracy would be of no import.

Secondly, it must be admitted that while there was a difference between the control group and the expert system users the mean of the expert system users was not so high as to cause elation--even with the expert system the experimental subjects averaged slightly less than two-thirds of the questions correctly answered.

The large gulf between plausibility and effectiveness must be bridged by future research if expert systems are to be used in an educational setting. Would an hour of inservice, or a short training manual with examples increase the performance of educators? Or are better, more explanatory or "help" frames in an expert system necessary? Would the subjects have done better with differently worded questions by the system, or does an expert system require at least a minimal level of knowledge on the part of the user, and if so how much?

In addition, another line of inquiry for future research is the problem of problem types; there does not now exist a taxonomy of problems, the solutions to which could be effectively modelled by an expert system. The present research modelled a statistical decision making process, a fairly well structured problem. Could less structured decision rules be as successfully modelled? Clearly, most educational decisions are of a fuzzy nature and situation specific. Therefore, a computerized "expert" may not be easily programmed in some situations. The present research has demonstrated the feasibility of an expert system in an educational context. Which educational problems are tractable through the use of expert systems and which are not? These answers must be supplied by future research.

REFERENCES

Hirsch, A. (1984, March). Artificial intelligence comes of age. Computers and Electronics, 63-67.

Olsen, C. R. (1984). Decision support for statistical hypothesis testing: An experiment. Unpublished doctoral dissertation, The University of Iowa, Iowa City, IA.

Turing, A. M. (1956). Can a machine think? In J. R. Newman (Ed.), The world of mathematics (pp. 2099-2123). New York: Simon and Schuster.

Webster, R., & Miner, L. (1982, January/February). Expert systems: Programming problem-solving. Technology, 2(1), 62-73.

Where Would You Go Tomorrow If Disaster Struck Today?

Charles G. Hadduck
Portland Public Schools
P.O. Box 3107
Portland, OR 97208

ABSTRACT

The possibility of disaster in the Data Processing center is a very serious subject. Many data processing centers continually run in a high-risk environment where good security is perceived as important but tends to be only incident oriented. Disaster/ Recovery planning is, however, the critical issue and goes beyond the technical aspects of physical security and disaster prevention. This presentation will investigate possibilities that a disaster could occur, perplexities that would ensue and the potential for an orderly recovery.

INTRODUCTION

Facing the issue of a possible disaster in the data processing center is a subject that most organizations would rather avoid. Over the past decades organizations have been relying more and more on the computer to store sensitive information. As the environment that utilizes the information also includes a wider range of people, the need for contingency planning has become increasingly evident. Only in the case of an actual disaster does there appear to be a justification for the investment in a Disaster/Recovery Plan. On the other hand, almost every organization will readily justify having fire insurance and other types of liability insurance to protect their physical assets. The emphasis of this paper is the equally important need for due consideration of the value of information assets and the corresponding value of a Disaster/ Recovery Plan.

DISASTER/RECOVERY PLANNING ISSUES

Perceptions of the need for Disaster/ Recovery planning often differs widely at the various management levels in an organization. Througout the data processin industry the issue has become very prominent in recent years, due primarily to the experiences of disaster in some well known companies.

As a technical issue, security is approached in a manner that tends to be incident oriented. The major focus is on physical security of the data processing facility, off-site security for important data files and technical security to prevent the system from being compromised. These concerns are, of course, extremely important. However, the efforts to provide an appropriate level of physical, off-site and technical security may be only cosmetic when one considers the broader aspects of the security issue. The real question is, if a disaster comes, has anyone bothered to decide what you are going to do, or who is going to do it, or how you are going to do it or where you are going to do it? These technical issues, it must be noted, focus primarily on disaster prevention issues.

As a management issue, security, to be effective, must be perceived as an integral part of the organization's management policies. There must be an awareness of the importance of a comprehensive planning effort, the value of the organizations information assets, the probability that a disaster could occur, the possibility for an orderly recovery and the consequences of being unprepared in event of a loss. There needs to be a rational approach to the security issue whereby company resources of personnel, time and budget are sufficiently allocated on an on-going basis to attempt to answer the questions of what, who, how and where mentioned above. Fortunately, there has been, in some areas, upper management which has become involved in the promotion and implementation of an extensive contingency planning effort.

DISASTER/RECOVERY PLANNING OPTIONS

A Computerworld report indicates that fewer than 7% of all companies that experience severe damage to their data processing operations are in business five years after the loss. The major reason lies in the absence of an active Disaster/Recovery plan which includes a back-up site. Consider some of the available options and/or consequences of back-up processing should a disaster occur:

Out of Business - Unless your business is relatively unimportant, this is the least desirable of all options.

Manual Processing - A possibility if transaction volume is low and in-house expertise is available.

Redundant System - A viable option for a large company with distributed processing capabilities. Allows for disaster recovery testing. However, the cost is very high and requires a high level of coordination to insure compatability.

Reciprocal Back-up Site Agreement - A viable option for a small batch oriented data processing facility. Usually is very inexpensive because of the mutual arrangements. Maintaining hardware and softwware compatability is an on-going important consideration.

Shell Site Facility - Provides assurance of long-term availability and does not impact any on-going operations. Provides a starting point for developing a recovery plan, and is economical because costs can be shared by multiple users. Major obstacle is the availability of hardware for installation on short notice. Without installed hardware, no recovery testing is available.

Equipped Site Facility - Probably the most desirable option but requires a substantial investment from involved parties. Allows for periodic testing of recovery plans and is immediately available should a disaster occur. Requires a standardized equipment configuration and an investment of people resources to manage the facility.

Thus, if Disaster/Recovery planning is properly valued by the organization, the logical outcome will be the application of human and financial resources towards an on-going consideration of the possibilities for continuing in business should a disaster occur. Where you would go tomorrow should a disaster occur today is totally contingent upon how you are planning today for the possibilities of tomorrow. Anything less than an immediate and planned response to a disaster has the potential of putting your company out of business, perhaps permanently!

DISASTER/RECOVERY PLANNING ACTIVITIES

People and planning are the two most important aspects of preparing an organization for security preparedness and Disaster/Recovery. The planning steps and detailed procedures must be comprehensive, practical, achievable, cost effective, and currently tailored to your organization.

The presence of a Disaster/Recovery Plan presupposes the possibility that an occurrence could render vital Data Processing resources functionally inoperative for some period of time. It assumes that one or more forms of disruption of D.P. operations can occur, that the period of disruption will be of sufficient duration as to incur potentials for loss, and that recovery to a workable platform of operations in the shortest possible time, followed by full recovery, is the most desirable object.

The purpose of a Disaster/Recovery plan is to attempt a pre-determination of the logical recovery process and to reduce the number of decisions which must be made under duress when, and if, an incident occurs. It is the basis upon which all action should be begun. It therefore should be included in training sessions of all personnel who could be affected by a contingency.

Important aspects of Disaster/Recovery planning in action would include at least the following activities:

Analyze Minimum Downtime Requirements Since Disaster/Recovery is not a "business as usual" situation, the data processing organization will be operating under emergency conditions. You will not be able to accomplish everything normally done during the usual operating day, thus the critical survival applications must be identified separately from non-critical applications.

Assign Staff Resources - It has been estimated that a minimum of 2 man-years effort is required to develop a comprehensive Disaster/Recovery plan. This assumes that an employee is assigned full time to the task. However, a simplified plan is better than no plan if a comprehensive effort cannot be authorized. The part-time assignment of one person can bring about a usable plan in a short period of time.

Actuate the Planning Process - The preliminary activities in the development of your plan should address what your company has to do to continue its fundamental operations! It is essential that you understand your data center production workload and on-line transaction volumes. Important considerations are as follows:

- Determine your essential records and equipment (be sure to include telephones and telecommunications).
- Determine where these records and equipment are available on a back-up basis.
- Determine your manpower requirements for Disaster/Recovery. Who are your key employees - those you need to get your business going again?
- Determine your space requiirements and layouts for recovery operations.
- Determine hardware availability and commitment of hardware/software vendors, telephone company and telecommunications vendors. Do not forget associated equipment that interfaces user applications with your mainframe; e.g., minicomputers, special process computers, terminals, etc.
- Determine your recovery site needs. A most important consideration will be the complexity of and dependance upon the data commmunications network of your organization.

As a conclusion of the above stated planning process, a remote backup site would be most desirable and provide the least number of obstacles to recovery from an incident of disaster and loss. Without a backup site as part of your plan, be it a shell site or equipped site, chances for survival from the loss of your data center are at best questionable. Ample evidence is available from companies which have experienced a disaster, and who have included a backup site in their Disaster/Recovery planning, that indicates an unquestionable value of having a remove backup site available.

DISASTER/RECOVERY PLANNING CONCLUSIONS

In 1979 an informal group of Honeywell medium and large scale users began to meet on a quarterly basis as the Pacific Northwest Contingency Group. The purpose of the group was to share mutual concerns and plans in the area of Disaster/Recovery. From this large group of some 20-25 organizations, another group began to emerge which committed to expend funds toward the establishment of a shell site.

This small group, hereinafter referred to as the shell site group, began to meet bi-monthly in order to address in more detail all the ramifications of locating and managing a shell site. Portland Public Schools became a charter member of the shell site group and began a concientious effort to develop a long-range Disaster/Recovery plan which included the backup facility.

A vacant site was available which was formerly used as the data processing center of one of the shell site group members. After some negotiation, an acceptable contract was developed and a lease was signed. The site has adequate air-conditioning, 220v power service, space that can accommodate up to 3 medium systems or 2 large systems, support area for staff and a vault for media storage. The area is access controlled through a combination lock on the door. Adequate phone trunks for voice and data transmission are available.

Although no disasters have occurred, a very successful test was recently held which provided valuable information to the shell site group members. A local company ordered a new Honeywell DPS8 system whose delivery date was several months ahead of the scheduled completion date for the facility in which it would be housed. A sub-lease was arranged to allow for temporary use of the shell site facility. The system arrived as expected, and was installed and running in test mode within 48 hours. The shell site electrical power service and air-conditioning were completely adequate for their needs. Four months later the system was moved to its permanent home and the shell site vacated, all taking place over a weekend. Thus, the reality of activating the site on short notice proved to be workable with minimum difficulty.

In conclusion, the shell site group continues to grow in knowledge and experience. The philosophy of maintaining the site as a cold site or moving towards an equipped hot site is constantly under review, and methodologies for marketing the site are being pursued. The management commitment of the participating members continues to be strong viewing the knowledge and experience already gained as a beneficial return on investment.

As time goes on, organizations' needs continue to change. The Disaster/Recovery plan, backup site needs and recovery requirements must be in constant review. A one-time activity is not enough--Disaster/Recovery planning is effective only in the context of a long-term commitment.

From Course Requests To Examination Schedule

Jim Osborne
North York Board of Education
5050 Yonge Street
North York, Ontario

On the following pages are descriptions of a series of related programs which run on CBM, APPLE and IBM PC. These should be of interest to guidance counsellors and timetablers. Beginning with the entry of student course requests, these programs will create a conflict matrix, provide common course request lists, recommend where to schedule all sections of all courses, provide simulated and final loading runs of entire school, schedule individual students, print student schedules, maintain class lists and home form lists, and create an examination schedule based on a matrix produced by the stored student schedules.

All three programs are menu driven and extremely user friendly. Unlike many of the spreadsheet and word processing programs, no special keys or characters need to be remembered. Instead, the user is prompted throughout all of the routines. At the time of writing, only the Master Schedule Builder and Student Scheduler programs have been written for the IBM. The descriptions below for these programs apply to the IBM versions. The Commodore and Apple versions are, however, very similar. These descriptions follow the steps one would need to take in order to obtain the desired scheduling results. It is assumed that the programs have already been installed with the school's specifications.

MASTER SCHEDULE BUILDER

The course codes for the various courses offered in the school calendar are keyed into the computer and stored in a file. The conflict matrix numbers are also stored in a file. These numbers can be typed into the computer from a mainframe computer listing or generated from a course request file.

A course request file can be created and maintained to hold for each student in the school, all of his/her requested courses. An additional character, (1 or 2), can be added to the student's course request to indicate that he/she wants the course during either the first or the second semester. From this file the conflict matrix can be generated and tally and common course request lists can be produced. The course request file can also be used with the STUDENT SCHEDULER program to perform simulated and final loading runs. The conflict matrix on file can be read and printed for all courses, all single section courses, or all single and double section courses.

Teacher and Room files are created next. These are required in order to keep track of which teachers and rooms have already been scheduled. At this time a teacher can be flagged as being unavailable to teach in certain timeslots because of office or some other type of duty.

By the time the next stage is reached, decisions regarding number of sections and staffing must be made. With this information in hand the number of sections is entered into the computer for each course. At the same time teacher and room restrictions can be added for individual sections. A section can also be pre-placed if for some reason it must go into a particular timeslot. A range of timeslots can also be set for each section so that if a section must be taught, for example, in the morning or in one semester or the other, the computer will schedule that section accordingly.

At this point the computer has stored in files, all of the information it needs to schedule the sections of each course. A command from the main menu sets the scheduling process in motion. Over the next couple of hours a master schedule is generated. The single sections are scheduled first and the computer works its way up to the multi-section courses. Based on the conflict matrix, the best timeslot is chosen for each section then a check is made to ensure that the teacher and room are available and, if not, the next best timeslot is chosen. At the end of the process two reports are produced. One lists, by timeslot, all of the courses scheduled. The other lists, by course and section, the timeslot and, if previously specified, the room and teacher.

If the resulting master schedule is not workable for reasons such as teacher load or unavailable rooms, changes in restrictions can be made and the scheduling process repeated as often as desired. A simulated loading run against the course request file can be performed using the program, STUDENT SCHEDULER.

STUDENT SCHEDULER

The master schedule is keyed into the computer and stored in a file. This takes about 2.5 hours for about 350 sections. The maximum number of sections per course is 16 and the maximum number of courses is 400. As the information is entered, class list files are also set up for each section to hold up to 50 students. For each section, the timeslot(s), maximum class size, teacher, and room are entered. A separate routine makes it easy to make changes to any of this data. In order for students to be scheduled into home forms (optional) it is necessary to type in the names of the home forms and associated course/sections.

Simulated and final loading runs can then be performed using the course request file from the the MASTER SCHEDULE BUILDER. As each student's requests are processed, a running total of those scheduled and unscheduled appears on the screen. For any unscheduled student, a print-out is produced showing the student's course requests. During final loading runs, student schedules are printed as the students are scheduled. Between scheduling runs a command can be given to reset all of the class sizes to zero.

A second scheduling mode handles individual students wanting completely new or changed schedules. This part is often handled by guidance counsellors with students present. The student's course requests are typed in (optionally adding a seventh character to the course code to indicate semester choice) and the computer allowed to schedule the student while trying to use the smallest sections but at the same time keeping class sizes balanced. Once a section has reached maximum size the computer will no longer schedule any student into it. Once a schedule is worked out, the computer's suggestion is displayed on the screen with a prompt to accept or reject the suggestion. Accepting the suggestion allows the class sizes to be immediately updated. Also a record is appended to a temporary file with all necessary information to generate add and delete notices for teachers and school board. Another main menu option reads this temporary file and uses the information to update student, class list, and home form list files. Before returning to the main menu the student's new schedule can either be immediately printed or stored on disk for printing later in the day.

If the student could not be scheduled or if the suggested schedule is rejected then a requested course can be dropped or replaced and another attempt made to schedule the student.

A third scheduling mode makes it possible to override the computer's choices and specify into which sections a student is to be scheduled. This is called 'manual' scheduling. This mode can be entered directly from the main menu or indirectly if the student can't be scheduled or the computer's suggestion is rejected. Manual scheduling makes it possible to add students to classes which have reached maximum size.

A fourth scheduling mode makes it possible to change an existing student's schedule. This can be accomplished two ways. If the student's schedule has been stored on the disk then it can be recalled by typing in a few characters of the student's surname. It is not necessary to know the student's file reference number since the computer will search the files fairly quickly by name alone, starting with the first letter of the surname. For example, if you were looking for the name Charlie Horse you could type in HOR and the computer would start searching through the H's until it found a match and prompt you to confirm that it has found the correct name. If not confirmed then the search continues until all H's have been examined. On the other hand, if you knew the student's file reference number assigned by the computer, the student's record could be located immediately. Once found the student's present schedule is displayed on the screen and the opportunity is presented to delete any or all of the student's courses. Then the computer either goes into 'computer' or 'manual' scheduling mode, depending on the user's choice, in order to add additional courses. If the student's schedule had not been stored on the disk then his/her present courses could be typed in and the computer would follow the same routine as if it had located the student's schedule on the disk.

Besides the printed reports mentioned above, the following additional lists can be generated.

- master schedule with class size
- home form list
- students in class lists
- alpha listing of all students in school
- students by home form

Editing options from the main menu make it easy to change any data already stored on file. For example, you can add at any time additional sections or entire courses, change spellings, home forms, class sizes etc.

EXAMINATION SCHEDULER

The minimum requirement for this program is a list of courses that need to be scheduled and a conflict matrix. If a mainframe printed conflict matrix is available then this program can stand alone. All you would need to do is type in these two lists and allow the computer to develop the exam schedule.

The program can be much more valuable however, if you have the files maintained by the STUDENT SCHEDULER program. The EXAMINATION SCHEDULER can read the student program files and create the conflict matrix itself, saving the data entry time. The only data entry required then is a list of the courses that need to be scheduled and the semester for which the schedule is to be created. This would take only a few minutes.

During the scheduling process the program sorts the conflict matrix by the number of courses with which each course has no conflicts. The first course is then chosen for the first group. Along with this course is taken the next course with which it does not conflict. Then the next course not conflicting with the first is examined to determine whether there are any conflicts with the second course and, if not, is included in the group. This process continues until the entire conflict matrix has been scanned and a listing of the courses in the group is generated. Then a second group is made up in the same manner as the first. This process continues until all courses have been scheduled into groups.

The end result is the combining of all the courses into the minimum number of groups possible for scheduling the examinations with zero conflicts. This may create a problem if a school is restricted to 5 days of exams with a maximum of 3 per day. Such a school must have a maximum of only 15 groups of courses. If, on the first scheduling run, the program produced 18 groups then some condensing would be necessary with some resulting conflicts. When the scheduling process is started, the number of acceptable conflicts is entered via the keyboard. If zero acceptable conflicts resulted in too many groups then a second run could be made with 1 or 2 conflicts acceptable. With each scheduling run the number of resulting conflicts in any group is calculated and printed by the computer. This process is repeated until the number of groups is acceptable. If the computer indicates that the number of conflicts within a group is 5, then the names of the 5 students can be determined by the common course request feature from the main menu. The course codes are typed in and the computer examines the class lists to determine which students have the courses in common.

Each of these programs is a stand-alone program. That is, it can be used independently of the others. However, each program can use files read or written by the other programs. Used together, the value of the programs is increased. The programs are available from Computer Microsoftware
53 Thorny Brae Dr.
Thornhill, Ontario
L3T 3G5.

Administrator "Homo-Optimus": A Diagnosis Of Change And Relationships Affected By Technology

Dr. Geogre Goldstein
Sewanhaka Central H.S. District
555 Ridge Road
Elmont, New York 11003

An overview of inter-relationships affected by the introduction of change in an administrative setting in public schools. An examination of concepts of human relations factor with regard to organizational behavior, defective adaptation and psychodynamics of role play. A discussion of behavior modification and the commandments affecting change by today's technological society. An exploration of the needs for school administrator adaptation to change in view of the stampede to excellence movement.

In discussing change and the necessity of examining the quality of a product, I offer the following: "The engine, the transmission and the wheels are made of greed, fear, and invidiousness, fueled by superpep, high-octane money and lubricated with hypocrisy. Interior upholstery is all hand sewn of the skins of children of the very poor. Justice and mercy, charity and peace are optional items at slight extra cost. The steering gear is newsprint powered by expediency, but not connected with the wheels, and finally, there are no brakes. However, the rearview mirror and the horn are covered by our lifetime guarantee." **Is this product education?**

George Bernard Shaw said the reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore, all progress depends on the unreasonable man. I will explore with you concepts of change, and how we are in a homo optimus state, whereby a rational educator seeks to maximize only those personal interests which the theory says that he or she must have. Objectively determined self-interest has taken theoretical precedence in our psychologically minded world. Management today, through various models, is totally unrealistic, if not downright wrong. In understanding a compilation of recent efforts to understand change, we will explore various forms of behavior in understanding the work environment of our schools.

We are now embarking on a society-wide effort to create excellent schools. What culture would ever aspire to mediocrity in education, or permit the primary mechanism of acculturation to fall into disrepair? Despite its present appeal, the excellence in education movement would have much more validity if it were a first. But, it is not. It is part of a series of well intended efforts to improve public education in America. Remember the excitement of creating innovative schools? How about State and Federal programs to reform public schools? Then the effective schools movement. And now, the stampede towards excellence.

Learning is the indispensable investment required for success in the information age we are now entering. A high level of shared education is essential to a free, democratic society, as written in A Nation at Risk: The Imperative for Educational Reform by the National Commission on Excellence in Education. Last year, major reports on the state of American education captured the nation's attention; these studies issued both a warning and a challenge. Our educational system is threatened by a decline in quality and achievement. In order to continue to thrive as a society, we must recommit ourselves to educational excellence as an essential national priority. One thing is certain. To be successful, everyone must be a partner in this effort: teachers, students, business, labor, and government.

The path to educational excellence must include technology and its changes affected by each succeeding generation of learned scientists and inventors. Our schools, for the most part, do not assimilate to change easily. Field agents such as district curriculum coordinators, consultants, State Education Agency staff, and intermediate service agency staff can often be the key factors separating success from failure in change. The same program can be highly successful in one school and miserably ineffective in another. Efforts to change, work in some places and not in others. There is a basic argument that existing school contextual conditions inevitably interweave with a change process to produce substantially different results from school-to-school and, yes, from community to community. Indeed, the effectiveness of administrative activities, how planning was carried out, the success of involving teachers in the process, how widely classroom changes were implemented, and how long the changes lasted are all acutely susceptible to the influence of certain contextual conditions today: (1) The availability of school resources; (2) the availability and nature of incentives for innovative behavior; (3) the nature of a school's organizational linkages; (4) existing school goals and priorities; (5) the nature and extent of faculty factions and tensions; (6) turnover in key administrative and faculty positions; (7) the use of knowledge and current instructional and administrative practices; and, (8) the legacy of prior changes.

William Shakespeare could not have anticipated the information revolution when he wrote, "The past is prologue." There is nothing in thousands of years of human history to prepare us for the incredible changes in our lives and our life styles that computer and communications technology will generate. Advances in data processing capability are the driving force behind the information revolution. I still wonder at the fact that today an individual can walk into a computer store and walk out with a machine that far surpasses, in power, speed, and operational convenience, the computers that the largest corporations were using back in the 1950's. I can think of few, if any, other fields in which there has been so much progress in such a short time.

By 1990, 3 out of 4 workers will be working on computers. We are rapidly approaching the time when a majority of all workers will have grown up in the computer era; to these individuals, the technology will not have the mystique associated with it that has characterized the attitude of older generations. Few school districts will prosper in the next decade unless they develop an overall strategy for acquisition, utilization, and management of information services for both administrative and student use. To accomplish this goal, administrators and information professionals must work together to master the current explosion of computer and communications systems. Now, as we move toward a recognition of the importance of hammering out a concept of information use in management, we must work to develop new ways of thinking that will lead us to both new sources and new ways to manage information in our schools. These strategies must be adequate to meet the promises of technology over the next decade to give us better accessibility to our information resources and allow us to mobilize information so that we can function competitively in our school programs.

Organizational Behavior

Some schools readily welcome new practices to improve student learning, while others seem impervious to the winds of change. Often it has been asked why are curriculum coordinators and specialists successful in some schools and fail so miserably in others. Indeed, changing urban schools is difficult and yet, too, surburban schools are not a piece of cake. Schools are not directed by a deus ex machina preoccupied with mere performance quotas. On the contrary, schools are made up of individuals, each of whom brings his own unique personality to bear on decision-making processes. Inevitably, the framing of decisions by individuals, given the complexity of individual personalities and the intricacies of group interaction, causes distortions. Yes, in short, the myth of organizational rationality needs to be re-examined in the context of what is known about the role of the unconscious in human motivation and its impact on decision-making. **Indeed, personalities can make or break any school system.** Other influences that have had an impact on whether or not computer technology should be brought into school are stress, time management, and psychological barriers. All too often administrators, as well as teachers, have had some psychological barrier to the impact of computer technology. Some, literally, are afraid of machines. They are afraid of pressing buttons, afraid of programming, and of using their time to learn new concepts. This, ultimately creates stress. I have found, in my years of educational experiences, that it is often easier to work with those on a clerical level who are learning a new task vis-a-vis administrators than tenured teachers who must make change in structure. One of the examples studied psychologically is that of the normal positive. Individuals who have shown a tendency to make change, do so basically because it's pressured or pushed on them. This normal positive and the interaction of irrational behavior can exist in many work organizations. Some schools may force people into irrational and pathological behavior. Various human passions may interact with work in the bureaucracy to produce either congruents (no symptoms), or incongruents (symptoms). Indeed, in every bureaucracy there exists some irrational behavior.

In many work organizations people fear change and in looking at irrational behavior, we could find that some people in organizations exist in their lust for power and glory and their desire to subjugate and/or destroy others just for the greed of personal gain. There also exists those who have passive dependency. Some people like to be browbeaten or pushed around just to be made to work and accept change as a form of masochistic humiliation.

Psychodynamics

Successful school administrators show high drive and achievement desire. They conceive of themselves as hard working, achieving persons who must accomplish in order to be happy. The areas in which they do their work are clearly different, but each feels this drive for accomplishment. Nowadays, many administrators thrive on the practice of making change, and not standing on past achievements of their predecessors. The idea of authority as a controlling but helpful relationship over peers, as well as subordinates, is another indication of many administrators' desires to achieve that which is different, to bring something new to that which was old. I can recall how an administrator sucessfully engineered the concept of ordering 50 computers into a school district. A year later he was replaced by a new administrator who decided that they were all wrong and inadequate; a Board's investment of $180,000 was thrown out the window. He proved that (1) he made an achievement; (2) he was an authority; and, (3) his need to prove himself created this interpersonal tanglement. His decisiveness had strong aggressive and underlying motivations.

Computer purchases often have less to do with enthusiasm than a sense of coercion or compulsion. A California group called the AD HOC Committee on Basic Skills Education argues that money spent on computers "could better be used in improving curriculum, teacher strengthening and greater basics emphasis." They characterize promotions of elementary school computer education as a merchandising scam designed to convince concerned educators that today's computerless child is tomorrow's dropout. There can be little doubt that mass marketing has played an important role in the adoption of the new technology. To understand the forces behind this movement, one must keep in mind the distinction between hardware and software that directs the system for everything from word processing to arcade games.

Seymour Papert writes in his influential book, MINDSTORMS, "in many schools today, the phrase, computer aided instruction, means making the computer teach the child." Perhaps one might say the computer is being used to program the child. Many boards of education are faced with the concept of change and, specifically, should they make technological change in the schools. In the magazine Future Computing, a market research firm estimates that schools spent about 20 million dollars on software last year, and that another 110 million dollars was spent on educational software for the home. With money like this at stake, it is not surprising that the forces of mass marketing are increasingly important. Perhaps our schools, prior to the end of this decade, will be in the mass marketing concept for computer education and, most notably, the concept of computer literacy.

Educators today are being bombarded with competing and often grandiose claims about the values of different approaches to computer based education. Controversy is widespread, and decisions with important policy implications are being made on the basis of truly minimal research evidence.

Commandments of Change:

1. Commandment number 1 of any truly civilized society is this - let people be different.
2. Once you've opened a can of worms, the only way to get them back in is to use a larger can.
3. The more ridiculous an idea is, the better its chances of catching on.
4. Careful planning and minute attention to detail never succeeded as well as dumb luck.
5. There is never time to do it right, there is always time to do it over.
6 Anything that has been put together will fall apart sooner or later.
7 Things will get worse before they get better, and nobody said things were going to get better.
8. Everyone has a great idea that won't work.
9. Any given program when finished is obsolete.
10. New systems generate new problems.
11. Computer technology is dominated by those who manage what they do not understand.
12. If contractors built buidings the way programmers write programs, then the first woodpecker that came along would destroy this place.

In thinking about change perhaps a quote by Marie Curie said it best. **"Nothing in life is to be feared. It is only to be misunderstood."**

REFERENCES

Diebold, J. "Managing Information - The Challenge and the Opportunity." American Management Association, AMACOM, 1985.

Kets, M. deVries. "The Irrational Executive." International Universities Press, 1984.

Mann, D., ed. "Making Change Happen." Teachers College Press, 1978.

A System And Strategy For Data Processing Planning For Universities

Dr. Donald L. Dawley
Miami University, Department of Decision Sciences
223 Culler Hall,
Oxford, OH 45056

Universities operate in an environment characterized by resource constraints and dynamic technological change. Effective data processing planning will assist schools to identify and acquire the needed faculty computer expertise, hardware, software, and facilities necessary to make effective use of computers for instruction and research. This paper provides discussions of one approach to effective data processing planning for universities.

THE NEED FOR PLANNING

Universities are complex organizations that have important roles to perform within our society. Goals are formulated to clarify the school's reason for existence and also to give direction to their activities. Schools, like most other organizations, operate under the constraint of scarce resources. Effective planning can significantly improve the efficiency of the organizational activities directed toward achieving those goals and increase the degree to which their goals are achieved.

PLANNING GOALS, GUIDELINES, AND STRATEGY

Goals provide the planning destination, guidelines provide the planning parameters, strategy provides clues to the best routing, the planning process identifies the route to take to best reach the goals, and the planning documents the collective wisdom and serves the function of roadmap. Goals, guidelines, and strategy are of critical importance if the objective is to obtain the highest rate of return from the resources employed.

Planning Goals

Most universities have carefully worded general statements of mission which are intended to establish the overall character and framework of the institution. These statements rightfully set the tone for the planning process. Fewer institutions have goals specifically relating to data processing and fewer still have DP plans. The DP goals and plans at lower levels should track with goals and plans propagated at the university level. A plausible DP goal for a university could be "for every graduate to be computer competent."

Universities need to define their goals in sufficient detail so that an educational program can be planned and resources acquired and organized which facilitate the educational process. One approach is to first formulate general goals and then break them down into more specific supporting goals. One example is as follows:

General Goal:
  Supporting Goals: Graduates are to be computer competent.
  Faculty: A critical mass of faculty members in each functional area must be

expert in the use of computers in their functional area.

Students: All must be able to make effective use of computers and software tools suitable for their discipline.

Facilities: Compatible and standardized systems must be available in faculty offices, classrooms, and labs.

Planning Guidelines

Planning guidelines can be derived from many sources including the mission of the university, good pedagogy, the environment, management considerations, and the sources and availability of sources. A few guidelines are provided for consideration.

1. Educational requirements should "drive" the plan, rather than technology.
2. Faculty expertise must be developed and maintained within each functional area.
3. Computer proficiency must be developed over time through education and experience. Regular use is necessary in order to retain proficiency. The plan must be pedagogically sound.
4. All of the functional areas must be involved in the planning process.
5. Existing computing resources should not be duplicated unnecessarily.
6. The computing requirement should be matched to the appropriate technology.
7. Hardware and software should be standardized where possible.
8. The plan should be consistent with the priorities and resources of the school.

It should be kept in mind that guidelines serve a positive function and must be used with good judgment. The number should be kept to the minimum requirement to guide planning activity. Too many guidelines can make creative planning very difficult. It should be kept in mind also that a guideline can be set aside when circumstances warrant.

Planning Strategy

It was stated earlier that strategy provides clues as to the best routing. The appropriate strategy is situational. Universities which have few computer resources and have been unable to make effective use of computers for instruction and research may initially want to adopt a strategy that gives priority to the education of faculty members. Universities will also have to give priority to the hardware, software, and facilities needed to support the necessary educational program. This strategy recognizes the lead time necessary to train faculty members and develop instructional materials before the use of computers can be introduced into the curricula. Universities that have substantial resources and effective computer use can develop a strategy that is geared to staying current with the technology and even providing a lab for faculty to research and experiment with state-of-the-art software, hardware, and systems.

A rationale strategy could be to create a plan that recognizes three progressive levels of computer skills necessary for faculty and students to be recognized as having achieved computer competence.

Skills levels:

Level I: Basic skills. Familiar with operation of equipment and software.

Level II: Experienced user who has the ability to use selected functional discipline software for analysis, problem solving, and creative activities.

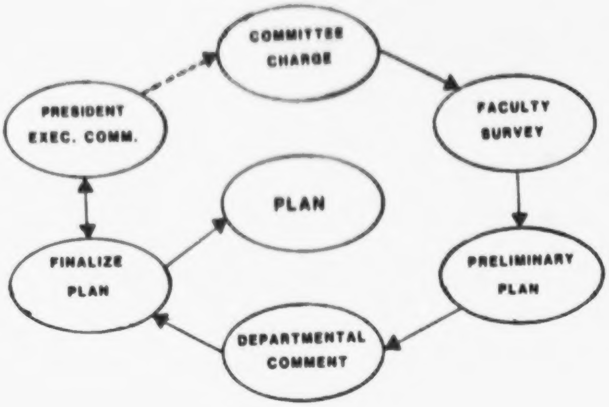
Level III: Proficient, current user with the minimum skill level needed for area. Considered computer competent.

1. Ability to use the computer as a tool with attention focused on functional activities.
2. Ability to use in a meaningful way, the computer in undergraduate and graduate courses.
3. Awareness of the availability and use of hardware and software in professional area.

THE PLANNING PROCESS

Planning in an educational institution, must have the strong support of top management. Without adequate support of administrators, the planning process is, or soon will be, a waste of time. Competent, interested, and respected faculty members must be appointed to do the DP planning. Failure to place valued faculty on the committee will result in either some planners doing a disproportionate share of the work or will result in a lower quality plan. It is also important that computer expertise be available to the committee either through appointment or in the advisory capacity. The planning procedure must be consistent with the organizational structure and the way decisions are made within the university. The president should initiate the planning process through the use of an appointment letter that identifies the planning committee members. The members should usually be appointed for one year at a time. However, a balance of returning and new members should be maintained in order to provide continuity and yet introduce new ideas and perspectives. The committee will progress more efficiently if the chair of the committee is a returning member. The president also needs to initiate a committee charge letter that clearly defines the responsibilities of the committee. The letter should be specific that the committee is advisory in nature. The committee can also function in an advisory capacity to review and make recommendations to the president on computer resource requests which are to be funded at the university level. The committee has the representation, computer expertise, and plan knowledge to evaluate the request with respect to SBA resources and consistency with the DP plan. The appoint and charge letters should be distributed to all faculty and staff members.

PLANNING PROCESS



The initial activities of the committee must be concerned with understanding its charge, reviewing available documentation, and formulating any changes to past goals, guidelines, and strategy. The committee should initiate selected surveys to determine future computing requirements within its first few weeks of operation. The surveys at a minimum should include all faculty members, selected students with emphasis on seniors, graduates, and employers. The committee also needs to brainstorm for requirements based on the projected future environment and their collective experiences in functional disciplines. The requirements should be organized by year in priority sequence and placed into a preliminary plan. Computer expertise should be brought to bear on the selection and timing of new computer hardware and software requirements. The rapid rate of technological developments often results in the superiority of hardware and software being a function of the timing of product announcements. A specific product is often superior for only a short period of time after its announcement. This leads to giving greater attention to:

1. Meeting required computing capability.
2. Present and potential use in business.
3. The quality and reliability of the vendor.
4. Initial cost and licensing arrangement.
5. Vendor's record of updating product.
6. Cost of upgrading installed inventory.
7. Quality of documentation and assistance provided.
8. Compatibility with installed resources and the needed mode of operation.
9. Selection based on the most advanced technology e.g.
   a. 32 bit vs. 16 bit processor or DMBS package
   b. 640K vs. 256 memory
10. Timing of purchases in situations where product announcements are known or can be anticipated ahead of time.

The estimated dollar costs should be provided for as many years as the institution budgets. The plan should be organized and presented so that members of the faculty and staff can understand it without a tutor. The plan should at least include:

1. Executive summary, table of contents, and foreword.
2. Introduction that provides the goals, guidelines, and strategy.
3. Faculty and student development plans.
4. Hardware, software, facility, and other computing resource requirements identified by year.
5. Detailed data in appendices.

The preliminary plan should be forwarded to the departments and major managers for review, comments, and suggestions. The responses received are a good indication as to whether planning is important at an institution. Planning is important if the responses indicate the plan was carefully read. Planning is not important if it is obvious that the plan was not read. The comments received should be carefully considered and changes made when appropriate. The finalized plan should be sent to the president for review and approval.

Concluding Thoughts

Information technology will increasingly provide the opportunity and challenge for educators and institutions to be more creative and effective. The intelligent use of this fast moving technology will increasingly require effective planning in order to make the knowledgeable curricula and resource decisions that must be made.

Managing Information: Security Access And Dissemination - A Conceptual Model And Case Study

Dr. Roy K. Bhagaloo, Director
Department of Systems and Data Processing
Muscogee County School District
Columbus, Georgia 31906

Many organizations are experiencing the impact of expanding computer activities, the concepts of the information age, corporate data base, and information as a corporate asset, as well as the escalation of data communication, and emphasis on information security, reliability, and availability. This paper attempts to accommodate this impact by presenting and relating an Information Management Model to an online approach currently in operation in the Muscogee County School District.

SECTION 1: INTRODUCTION

There is discernible in the study of computer utilization, a recent tendency toward a rapid expansion of computer resources in educational and other organizations, reflecting a strong desire to utilize these resources. However, affiliated with this tendency and its subsequent desire for utilization, are certain interrelated factors which have presently positioned themselves in the forefront of computer utilization, and will continue to occupy that position in the future as well. Among these factors is the concept of the information age, which acknowledges the increasingly important role played by information in the management and decision-making process of any organization. This acknowledgement of information's important role represents a logical extension and evolution of the next factor, which postulates that information be viewed as a corporate asset and be treated as such.

Perhaps the most significant factor relating to computer utilization, vis-a-vis the availability and dissemination of information, is the third factor embodying the concept of the corporate data base. Essentially, the corporate data base concept views information as being "owned" by the organization as a whole, as opposed to any particular operating unit, and calls for a central apparatus to serve as a depository and dissemination unit for all of the organization's information.

The fourth factor is a perception that the role of data communication is escalating and becoming increasingly vital in disseminating information throughout an organization. Indeed, data communication, whether local or remote, becomes one of the most important vehicles for making information available to decision-makers pursuant to the concepts of the information age and corporate data base.

The fifth and final factor assumes the existence of information security and integrity, not only from the viewpoint of information being a corporate asset and, as such, should be secured and reliable, but also from the point of view that information "boundaries" need to be established in order to ensure that the corporate data base concept does not evolve as an "all-or-nothing" operation, and that the "right" information is being disseminated to the "right" people. In other words, this final concept proposes that the organization's information be secure, reliable and available to satisfy job requirements.

In considering these interrelated factors, it can be substantiated that a need exists to conceptualize a model to assist in the development of an online information management system encompassing the accessing, disseminating, and securing of information in a rational and forthright manner, in order to optimize and enhance computer utilization. This paper addresses this need.

SECTION 2: THE MODEL

The conceptual model presented in this paper is limited to an identification and description of its objectives, components, and operational considerations and is described as the Information Management System.

2.1 SYSTEM OBJECTIVES

The objectives of the system are to facilitate the use of, and promote access to, the institution's data base while limiting such access to authorized users. Furthermore, these objectives include methods to ensure the data base's security, monitor and track the use of the data base, and enhance accountability related to all aspects of managing information.

2.2 SYSTEM COMPONENTS

The components which are necessary and constitute the basic ingredients for developing the system are identified and described in the following paragraphs:

2.2.1 Data Base and Screen Format

The first component relates to the design and development of the data base and screen format. The layout of the screens should be designed in a hierarchical or "menu-driven" format, with the "menu" being the lowest hierarchical level and broken out into meaningful and functionally self-contained data or information groupings.

In this way each menu could be portrayed on one screen and be limited to users in that particular functional operation who may require the data or information. Furthermore, care should be taken when designing data menu screens to ensure that meaningful and clearly-written instructions are displayed on the screens and that the function keys are used in a consistent manner. Also, screen design and format should include the capability of moving from one menu to another by using function keys while viewing any data menu screen.

2.2.2 Location/Operating Unit Numbering Scheme

The next component calls for the development and establishment of a location/operating unit numbering scheme for the entire institution. In order for this numbering scheme to be useful in managing information access, it should be devised in such a way as to accommodate individual units as well as summary groupings of related units. Consequently, a school principal's access to information could be limited to his/her school only, while the supervisor for elementary education could be limited to the group category level for all elementary schools by using one location number regardless of the number of elementary schools in the school district.

It should be borne in mind that the location/operating unit scheme should be developed to be constant and consistent so that it could be used as the location reference for the entire institution. It should be designed to accommodate expansion but should not be subject to regular changes. Furthermore, if an institution currently maintains several numbers for the same location, a standard location table should be developed to relate each of the different numbers to the standard location number, rather than mandating that all users change their numbering scheme and data files according to the new standard location numbering scheme. This method would forestall most of the opposition from the users and is another way of making the computer work for everyone.

2.2.3 Terminal Numbering Scheme

The third component deals with the development and establishment of a terminal numbering scheme in order to identify and monitor each terminal being used for accessing information (terminals in this model refer to devices which are used for display and update purposes i.e. cathode ray tubes or teletype devices). The numbering scheme should accommodate the establishment of a unique number for each terminal by combining the location/operating unit code with a sequential number for each terminal within the location/operating unit. This method would not only permit the identification of each terminal but also all terminals located at a particular location/operating unit. It should be pointed out that in order to facilitate the development and implementation of the information management system and to avoid redundancy of effort, the location/operating unit code used in the terminal numbering scheme should be the same as the code in the location/operating unit numbering scheme, if possible (usually determined by system software limitations). If this is not possible, then the location/operating unit codes in the terminal numbering scheme should parallel those in the location/ operating unit numbering scheme.

2.2.4 Password (Access Authorization Code)

The next component is the password, an alphanumeric code in which are embedded all of the necessary variables for accessing information. The first variable is the data base. The password should embed the data base access capability in order to permit limiting information access to the lowest hierarchical level or menu in the data base and screen format. In addition, embedding should permit access ranging from a high of all indexes to a low of one menu of one index. The maximum number of menus to be embedded should be determined at an early stage in the design of the system in order to prevent the occurrence of constant changes to the system during operation.

The next variable calls for embedding the type of access capability in order to permit the reading or writing (updating) of data/information at the menu level, and also permit this activity to range from a high of all indexes to a low of one menu of one index.

The third variable represents embedding the location/operating unit access capability so as to permit access limitation to the lowest hierarchical level of the location numbering scheme i.e. an individual department or operating unit within a location. The system should be designed in such a way as to accommodate a predetermined maximum number of hierarchical location codes rather than a maximum number of individual department or operating unit codes. In this way, the system would not have to be "expanded" as often.

Finally, the password should identify the person affiliated with it by interfacing with the necessary personnel files to display such information as the person's name, location, title and other such biographic/demographic type of data as needed.

2.2.5 Monitoring and Tracking Scheme

The final component of the Information Management System is the monitoring and tracking scheme. This component calls for monitoring and tracking in three specific areas which are identified and described below.

The first area calls for monitoring and tracking each terminal in the information network so as to identify and track the passwords (persons) using the terminal, the security violations occurring at the terminal, the data base usage (display and update) and the total terminal usage.

The second area deals with monitoring and tracking the data base in order to determine the passwords (persons) accessing each menu, the security violations of each menu, the total data base usage, and the updating of the data base through "journaling". The journaling aspect should be designed to present a "before" and "after" image of each data element being updated, the password (person) doing the updating, the date and time of the update, and the terminal used for updating.

The third and final area pertains to the monitoring and tracking of each password (person) to determine the terminals used, data base accessed and updated, and the security violations attempted by the password (person).

In this way, the total data base, terminal usage, and security violations of each password (person) could be ascertained.

2.3 SYSTEM OPERATIONS AND PROCEDURES

The major categories of procedures and operations to implement and manage the Information Management System are identified as follows:

2.3.1 Initiating Procedures

The user normally initiates the procedures by requesting access (read or write) to information in the form of a memo or other document after being informed of the contents of each menu through necessary in-house training activities. Eventually, the request for access is forwarded to the data processing unit after it has been processed through the institution's management approval cycle.

2.3.2 User/Data Processing Procedures

The data processing unit receives the approved request, contacts the users or other personnel for clarification, if necessary, and issues a password with the necessary embedded limitations. The password is then sent to the individual user in a security-conscious manner.

2.3.3 Automated Procedures (Internal)

When the user enters his/her password in the system to access information, the system automatically reads the password to determine if the limitations embedded in the password are commensurate with the information screen being accessed. If the limitations are correct, the information screen is presented to the user, if they are not, a security violation message is displayed and presented to the user.

In addition, the system automatically captures the password, terminal, data base menu and other information according to the monitoring and tracking scheme established.

2.3.4 Online Management

The online management procedures and operations include the types of activities and information required to manage information access in a timely manner. This aspect of the Information Management System is designed in such a way as to utilize the computer in doing many of the time-consuming manual activities and includes most, if not all, of the entire system operation.

First of all, the information tables required for password embedding and development is maintained online so that the individual password could be created and "tabled" in this manner. Next, specific screen menus are developed and used to verify access authorization of an individual, display alphabetic listings of all individuals with authorization for a particular menu or index, location, and other data. In addition, the monitoring and tracking information is, of necessity, captured online because of the method of logging system use on an occurring basis. Perhaps the most beneficial use of the computer, in this management process, is its ability to display to each user only the data or information menu screens which the user's password can access. In this way the user would not be tempted to violate security because the other data or information screens would not be seen, and for all practical purposes, be absent from the system as far as the user is concerned.

2.3.5 Reporting Capabilities

The system is designed to produce periodic reports whenever they are needed. The reports are generally for "journaling" purposes, whereby they are sent to users for reference, and also usage purposes to assist in the reallocation of resources to needs.

Linking Microcomputers To Mainframes For Period Attendance And Local School Flexibility

Wayne Goates
Greater Albany Public Schools
1130 W. Queen
Albany, OR 97321

With the recent developments in both hardware and software it is now possible for microcomputers and mainframes to share information and thus eliminate dual entry of student and course data. This article will explain the approach taken by two school districts to maximize the flexibility of local school microcomputer systems while providing an easy interface to a larger service district mainframe.

Most school districts make use of large mainframe or mini computers to store and process information relating to their students, course offerings and report cards. There is, however, a significant amount of information required at the building level (like period attendance, lockers, progress reports, etc.) that is generally considered too taxing or time consuming for large time-sharing computers. Until recently the use of microcomputers in the local buildings required the re-entry of information that already existed on the larger central computer. Advances in both hardware and software have now made it possible to easily transfer information between mainframes and microcomputers and thus eliminate the double entry of student and course information.

Two school districts taking advantage of these developments are the Greater Albany Public Schools and Lebanon Union High School District of Albany and Lebanon, Oregon, respectively. During the summer of 1984 these two school districts contracted with a local consulting firm to design and implement a computer system that would maximize the flexibility for local school attendance and student information on a microcomputer, while at the same time allowing information to be easily passed between the local schools and the IBM System/38 that serves the districts from the county Education Service District office.

The Education Service District and three high schools participating in the project each purchased an IBM XT computer with 256K RAM and 10 MB of hard disk storage. A 5250 Emulation card was also installed in each microcomputer to allow schools the opportunity of using their microcomputers either as stand alone systems for local processing or as terminals to the System/38. The IBM System/38 Transfer Facility software was purchased making it possible to pull designated student and course information from the mainframe at the service district office and transfer it to a floppy diskette readable by the IBM XT. This software package allows the user to quickly define, save or modify the data to be transferred and makes the conversion from EBCDIC files on the mainframe to ASCII text, DOS fixed-length random, BASIC sequential, BASIC random, or DIF files for the IBM microcomputers. While running the transfer program the user is prompted to define: First - the name of the System/38 file from which records are to be transferred, Second - the fields in the specified record format being transferred, Third - the order in which those fields are to be placed, Fourth - the conditions which test each record to determine if the record is to be transferred, Fifth - the order in which the transferred records will be sorted, Sixth - the name of the file on the microcomputer floppy diskette to receive the records.

To provide the most flexibility, dBASE III, a powerful microcomputer database program from Ashton-Tate, was used on the individual high school microcomputers to import and maintain the student and course information sent over from the System/38. With an extra 256K of RAM installed in the IBM XTs it has been possible to write a completely menu driven program in dBASE III that can perform any DOS command (like backup, formatting diskettes, etc.); run any word processor (allowing integration with the student attendance information); switch immediately to a terminal on the System/38; or run programs to produce reports either in dBASE III or other languages without ever having to leave and reload the dBASE III program. dBASE III also provides a very helpful report generator that leads administrators and secretaries through the process of generating their own reports.

Over eighty programs have been written by the consulting firm to produce desired student, teacher, and attendance reports. These programs are being marketed by Hi-Med Systems, Inc of Salem, Oregon as the Student PASS (Period Attendance for Secondary Schools). Since the information is stored in a database, local school personnel are able to modify their data and generate their own reports without having to pay a programmer for every desired customization. The student information is also available for use in other local school computer environments without having to be retyped. For example, one of the high schools in the project makes a copy of student information including class schedules and uses this information on the library microcomputer to keep track of students checking out books. The time saved by not having to enter student data and schedules is significant.

By using their own microcomputers, the individual schools are able to do attendance searches any time and pass the pertinent student and parent information on for word processing without having to worry about processing schedules at the central office. Because there is no competition for computing time individual schools can process their own daily attendance and print very detailed reports throughout the day. Among the more effective reports are the Teacher Unexplained Report which identifies for each teacher only those students who missed that teacher's class without a valid excuse. Such daily reports can become very time consuming if done on a time sharing machine for a few high schools having over 100 teachers, but are easily produced on a microcomputer within an hour after school is dismissed and available to be placed in teacher's mail boxes.

There is a great deal of third party support and off the shelf software available for individual schools when they are using their own microcomputers. Schools who make use of an optical scanner or bar code reader to enter attendance information will find a number of programs that can also make use of their input device for other purposes such as grading tests, doing inventory, etc.

Currently, work is being done to pass attendance information back to the System/38 to be printed on student report cards. (Teachers previously have had to bubble attendance information when they filled out the report cards.) By the spring of 1985, the schools involved hope to be able to transfer update information from the System/38 over the telephone (with programs like dBASE/answer) rather than using floppy diskette.

The project has been very successful. Local school adminstrators and teachers are provided with very timely, detailed and flexible reports without increasing the strain on the service district's mainframe or requiring dual entry of information.

Local administrators also appreciate the ability to generate their own reports without having to go through the exhaustive and time consuming committee process required on time sharing computers. It has been impressive to see how much has been developed that was not even considered in the initial specification of the system and realize that there are very few limitations for continued development.

A System For Computer-Managed Instruction

Charles G. Hadduck
Portland Public Schools
P.O. Box 3107
Portland, OR 97208

ABSTRACT

As continuous progress curriculum has been adopted by school boards throughout the country, a critical need has been expressed for a record-keeping tool to assist educators in the management of instruction; in determining student proficiency of level objectives; and, to more efficiently group students for optimum instructional effectiveness. This paper will describe the development and implementation of a computerized instructional management system which links level objectives, test items, test units, test answer keys, prescriptions, and other student information in a microcomputer data base under the control of the local school.

INTRODUCTION

Computer-Managed Instruction (CMI), as its name implies, is a tool within the scope of Continuous Progress or Goal Based instruction. This instructional process can be defined as a planned effort in a classroom to provide an effective program of formal and informal instruction, assessment, record keeping, replanning and monitoring of individual students. Since children learn and achieve at different rates and across differing sequences of skills, goal-based curriculum provides the teacher with a sequence of instructional objectives and a choice of instructional materials which contribute to the continuity of individual student achievement.

Over the past decade, goal-based instruction has become increasingly more prominent. The manual record keeping tasks of assembling, analyzing, and utilizing the student progress information, however require a significant amount of time. With the capabilities of the micro-computer now available to most schools, these record keeping tasks are potentially much more manageable. In addition, a wide range of other benefits are made possible which support and enhance the instructional management process.

At Portland Public Schools, a system for Computer-Managed Instruction had been mandated by the School Board to be utiized by a variety of curriculum disciplines, providing a record-keeping tool to assist edcuators in using the Curriculum Scope and Sequence, in planning classroom instruction, in determining student proficiency of level objectives, in assessing individual student progress, and in grouping students for optimum instructional effectiveness. A number of CMI packages are presently available from curriculum vendors and school districts. Several of these were reviewed in some detail. Though each was found to be quite satisfactory within the scope of their design, the uniqueness of the District's needs would requiire extensive modification to any of the systems available in the marketplace. In evaluating all of the ramifications of modification and adaptation versus in-house design and implementation, the latter was chosen because it offered the most versatility and in the long range, the most cost effectiveness. To this end, a CMI development project was established and a coordinating committee of educators, curriculum specialists and system development specialists was brought into being.

ORGANIZATION

As the Curriculum and Data Processing

Departments began discussions to accomplish the above goals, two major issues emerged which would require a unique cooperation between the two departments. First was the adaptation of a numbering structure to the Curriculum Scope and Sequence which was adequately comprehensive for current as well as future needs. In a relatively short period of time a numbering scheme was devised, applied, revised several times and ultimately approved. The initial discipline considered for the CMI system was the Reading curriculum. Subsequent applications in the near future will include Math and Science. To date, the numbering structure appears to be adequate in its adaptation across the several curriculum disciplines.

The second major issue was the implementation and training procedures for the system users. Proposals had been solicited from schools interested in CMI and nine were selected to be the pilot schools during the 1984-85 school year. A school teacher was transferred to the Data Processing Department as an office automation and training specialist for microcomputers. In cooperation with the Computer Curriculum Coordinator, introductory and user manuals were developed for the system. The D.P. training specialist also worked very closely with the system developers providing insights from teaching experience to make the product a usable tool in the school and classroom environment.

A third major issue, primarily for Data Processing, consisted of determining which hardware and software to utilize for the system. In keeping with the District's policy of utilizing a common microcomputer configuration of hardware for office automation and administrative processing, the IBM-PC/XT and the NCS3000 scanner were selected. The microcomputer software selected was DATAFLEX ™ from the Data Access Corporation. Connection to the District's host Honeywell mainframe was established via 1200 BPS dedicated phone lines, using either synchronous or asynchronous transmission protocol, depending on the particular school environment.

IMPLEMENTATION

The CMI system was designed consisting of two essential parts; the curriculum and student data bases which reside on the Honeywell mainframe, and the local school microcomputer system.

The Portland Public Schools comprehensive student data base has been in service for several years and is maintained either by batched input or online direct update. Since the Curriculum Department develops a standardized district-wide curriculum, it was determined that their data base would also reside on the central district computer and be maintained by the Curriculum Department.

The school based microcomputer system has been centrally developed and is transported via diskette to the schools as they implement the CMI system. With the hardware and software operational in the local school, the curriculum data is downloaded from the central Honeywell database and a local school resident program populates the DATAFLEX ™ database. This transpires usually once a year or if there is a significant addition or modification to the curriculum materials.

As hardware became available and was installed in the schools, the training specialist would meet with school staff to introduce and implement the system, providing the school as much assistance as necessary for them to use CMI operationally. Implementation has gone exceptionally smooth due to the extensive planning and the individualized training provided by the training specialist.

On a periodic basis, student/class data is also downloaded to the local school microcomputer. Most schools maintain their student/class assignments as part of the student information system, thus relieving them of a duplication of effort.

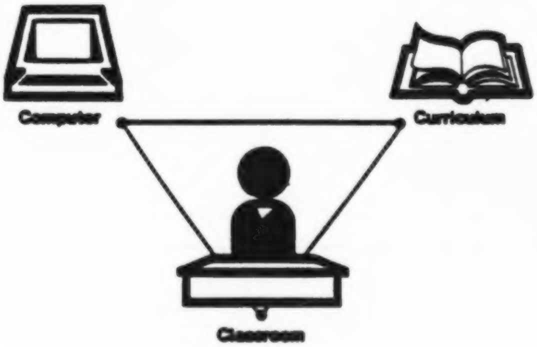
APPLICATION

At the local school, once the DATAFLEX ™ database is populated, there is adequate flexibility to adjust and/or regroup students as needed. This is accomplished via menu-oriented screens designed for ease of use. Three options are available: Teacher/Class assignment update; Class update; Student/Class assignment update. When all files are present and organized satisfactorily, the CMI system is now ready to function as the instructional management tool for which it was designed.

Instructional components include planning, direct instruction and practice. When asessments are required, tests are selected from the Curriculum Guide and administered utilizing a specially designed CMI scan sheet for marking the answers. Answer sheets are taken to the office where they are processed through the NCS3000 scanner. Immediately available are two management reports: An Item Verification Report and a Score Report. From these reports the instructor has valuable information in hand concerning the progress towards a proficiency of level objectives tested that very day. Information from these reports becomes valuable input in determining whether reteaching is required, regrouping of students is necessary or whether to continue the instructional process as planned.

Another report available on request is the Student Progress report. This will provide a year-to-date record of an individual student's progress, or lack thereof, towards all level objectives tested. At the end of a school year, or if a student transfers to another school, a printed copy of the Student Progress report is placed in the student's record folder as background information for future instructional level placement. Additional reports are under consideration and will be implemented upon a consensus from the principals of the pilot schools.

In conclusion, the goals of this system for Computer-Managed Instruction are being met and enthusiasm is growing. Additional outcomes envisioned are that teachers might know more about the progress of students, the implementation of curriculum specifications, the interpretation of classroom test results, and the most productive use of instructional materials. The most important outcome is a teacher's appreciation for the uniqueness of students' learning rates, styles and achievement levels.



Getting To The Core Of The Apple: Global Computer Development

David A. Kreutz
Elmont Memorial High School
555 Ridge Road
Elmont New York 11003

Getting to the Core of the Apple: Global Computer Development

This seminar presents a realistic approach to the development of faculty support for, and use of, the computer in all disciplines. Specific examination will be made of the steps necessary to create and maintain positive morale in the face of the need for change. Improvement of basic teaching techniques with the computer serving the tutorial needs of students will be addressed. Emphasis is placed on individualized instruction, learning attitudes and reinforcement of learning concepts in all disciplines affected by technology.

"Step right up ladies and gentlemen. Get your ticket into the 21st Century. No waiting, pushing, or special knowledge needed to understand what the computer can do for you and yours." Old fashioned approach to selling an idea? Too simple a statement about a complex piece of machinery? Not at all! Let's face it; one of the most common statements made about educators is that their teaching methods and acceptance of what is new is literally "old fashioned."

In reality, today's educators have the opportunity to take part in a universal educational revolution as significant as the development of the "electric viewing machines." These, of course, were the filmstrip projector, the overhead projector and the opaque projector. Still basic to every audiovisual budget in school districts throughout the world, these machines, in the 1920's, were not only common to schools but also were used by the educators in those schools. It was obvious that "Seeing is believing" and "A picture is worth a thousand words." Instruction was no longer limited to the sights and sounds of the teacher's experience.

Eric Asby, the British technologist, speaks of three great revolutions in human education: the forming of schools, the use of the written medium and the invention of books. The fourth revolution is the improvement in learning to be effected by modern technology, and particularly computers. "Particularly computers", makes the statement especially noteworthy since technology that has been available to educators since the picture machines simply has not been maintained as significant learning tools in the school. For example, the language lab has come and gone, as has the hand held calculator. Despite the fact that calculators have been available in a functional form for almost ten years, the preliminary findings recently released by the Second International Association for the Evaluation of Educational Achievement (IEA), indicates that calculators had found their way into few eighth grade mathematics classrooms. Yet, here is the computer, a technological development that resulted from the demands of military leaders, not educators, that has become a universal expenditure commonly found in the local school budget. What then is the educator doing to make that expenditure worthwhile in the learning process?

First, it is essential for the administrator to accept responsibility for providing an "action plan" for the acquisition and use of the computer in the school for students and faculty. Support for this involvement must be generated by the identification and use of colleagues and students who know how to utilize the computer. At the faculty meeting and the individual departmental meeting, the computer should be demonstrated and discussed. Specific information should be presented to the staff before the meetings so that time will be effectively spent in reassuring colleagues that the computer will make possible more time on task. PTSA representatives should also be present, and at the first opportunity, demonstrations of the computer should be made at the PTSA monthly meeting. These activities will generate continued support for future expenditures by the Board of Education in the area of computer technology. Expenditures must include education of teachers as well as students.

Not surprisingly, computer education has become a major part of the mathematics curriculum in most districts. This isolated exposure has turned off all but the most aggressive colleagues in the other disciplines. On the other hand, the haphazard approach to the introduction of the computer in school districts reveals that little effort is needed to find neighboring districts that have computer programs in science, social studies, English and language. Members of the staff should have the opportunity to visit these schools to see first hand the potential the computer provides for better instruction and learning outcomes.

The implications for partnership between the curriculum and technology are endless. The already antiquated approach to a curriculum that provides for a program of technology education separate from the other disciplines should not continue. The interdisciplinary approach to learning has been for years considered educator's mumble jumble. Few consider realistically the exciting learning outcomes that can result from true interaction between curriculum areas. Impracticality, the chief stumbling block, can easily be overcome by the simplest computer program.

Reading across curriculum lines would be a most natural application of the computer technology. The development of programs that tie writing skills to the application of scientific or mathematical concepts can assure learning at much higher levels of achievement. Social studies would benefit from the graphs developed in mathematics, while the teacher of art could demonstrate the beauty, excitement and form of art as examined through the analysis of a computer program. The possibilities are endless if the educator is willing to become the student.

Inservice computer programs for colleagues should be commonplace. These programs must be "inviting" in nature. A reasonable time schedule, credit incentives and the opportunity for practice should be available. The latter point must include the opportunity to use a computer during the preparation period, after school and in the evening. During the lengthy vacation periods throughout the year, teachers should have the opportunity to take the computer home. Realistically, it provides the educator with the instrument to practice, while also preventing the theft of the computer from an empty school.

A common element of the "action plan" must be attendance at educational fairs and visitations by the business community to the school to see the computer's potential. Don't wait for colleagues to "volunteer" to take part. Specific invitations should be made to the members of the staff who all to often don't become involved. Their lack of involvement over a period of time makes them a teaching liability.

Essential to the success of a total computer program is promoting the obvious reasons why technology is an intricate part of today's school.

Attractive, clear, specific materials should be distributed regularly to all the staff indicating the advantages of utilizing computer technology for instruction. This includes the following: individualized instruction, constant monitoring of students progress, improved student attitudes towards learning, less formal teaching and more learning, interactive learning, and better use of the professional staff. Secondary level teachers, as well as many elementary school teachers have students in their classes who are significantly skilled in the use of computers. How then are the educators using this knowledge and skill for the improvement of the learning process?

Exercises to reinforce class instruction can be purchased or developed by the teacher. Students who fall behind in basic understanding of content can be offered individual remedial activity. Rearranging questions and numbers on tests will give a teacher different tests for different classes. This can also be used as make-up tests for absent students. Learning tasks can be offered at the right level, thus avoiding frustration. As students master knowledge, they are able to advance while the slower students can be offered endless repetition without embarrassment.

John Dewey stressed "learning by doing." The computer offers this opportunity where the VCR and TV monitor do not. Teachers now will find that many of the frustrations that were time consuming, such as creation of tests, quizzes and homework assignments, can be stored in a "departmental bank." All colleagues will interact in development of materials for use in specific grades and ability levels. The curriculum can be more faithfully followed and departmental exams at the end of the year will be valid, consistent and reliable measurements of learning outcomes.

Administrators and supervisors must accept the responsibility to make every teacher a participant in this revolution.

REFERENCES

Markert, L. "Technology Education: Its Place and Importance in the Curriculum." NASSP Bulletin, May 1984.

McCrosky, C. "The Administration of English in the High School Curriculum." English Journal 7, February 1984.

Ramer, B. "Computer Technology: Be a Leader." SAANYS Journal, New York, Vol. 13 No. 1, Summer 1982.

Travers, K. "Mathematics Achievement in U.S. Schools: Preliminary Findings From the Second IEA Mathematics Study." Phi Delta Kappan, Bloomington, IN., February 1985.

Planning Computer Labs

Lois Edwards
Minnesota Educational Computing Corporation
3490 N. Lexington
St. Paul, MN 55112

This paper consists of excerpts from the first chapter of MECC's Computer Lab Guidebook, which is scheduled for publication by summer 1985. Other chapters cover staffing the lab, selecting software and hardware, organizing the environment, and operating the lab.

The District Technology Plan

Computer labs are a relatively new phenomenon in schools. They should be viewed as part of a comprehensive district plan for integrating technology into education. A computer lab can support the goals of a district technology plan by helping to incorporate computer activities into the curriculum. The lab should be designed to meet the instructional needs outlined in the technology plan.

Creating Computer Labs

The process of planning and operating an effective computer lab involves a unique combination of instructional and technical considerations. These include specialized instructional principles, such as criteria for evaluating software; technical information, such as the specifications of computer hardware; and management concerns, such as scheduling and staffing. Careful attention must be paid to a myriad of details ranging from software storage to lab security. The Computer Lab Guidebook organizes this information in a systematic way and provides a practical approach to planning and operating a computer lab.

Advantages Of Computer Labs

Computer labs have come into existence as schools have been able to purchase computers in greater numbers. When computers were more costly and schools owned only one or two computers, there was little need for computer labs. Now that interest in educational computing is increasing and the price of computers is dropping, schools are purchasing computers in greater numbers. At this point, questions arise concerning how to best allocate this valuable equipment. Should one computer be assigned to each classroom? Should the computers be placed on moveable carts and taken to classrooms as needed? Should they be centralized in one permanent location where all students and staff have access to them?

Many schools have found that centralizing their computers in a lab setting results in the most effective use of their equipment. In addition to operating a lab, however, most schools decide to place several computers on portable carts for classroom use. Teachers can reserve these "floaters" and have them moved to their classroom for the day's computer activities. Some schools prefer to keep their computers on portable carts, but group them in a lab four days a week and schedule them as floaters on one day.

Although classroom computer use can also be effective, there are several advantages to placing computers in a lab.

Helps Meet Curriculum Goals

A primary reason for creating computer labs is their effectiveness in meeting curriculum objectives, both in computer literacy and in subject areas. Grouping twelve to sixteen computers in one location enables an entire class to participate in an instructional activity at the same time. Teachers can more easily incorporate computer activities into their lesson plans since all students can participate in the activity during the same class period and since all the necessary software and other instructional materials will be close at hand. In addition, cooperative learning objectives are met through the interaction of students working in pairs at the computers.

Some instructional computing activities are difficult to carry out without a lab setting, for example, Logo, keyboarding, and word processing. Applications such as word processing and some individualized drill and tutorial courseware are best offered in a lab with one student at each computer.

Encourages Computer Use

Many schools have observed that computer labs encourage more teachers to use computers in instruction. Teachers often prefer to take the entire class to the lab rather than find ways to use one or two computers in the classroom. Teachers seem to regard the computer lab as evidence of their school's committment to computers in education. This is often an incentive for them to begin using computers. The support provided by the computer lab coordinator or aide can also be instrumental in encouraging teachers to use computers.

Makes Efficient Use of Hardware

The lab environment permits more efficient use of the hardware. A computer lab located in a separate room can be open to students and staff at the end of the day. Afternnon, evening, and weekend use of computer labs is especially valuable at the secondary and higher education levels. Community education groups can often make effective use of labs outside of regular school hours.

Labs also facilitate the sharing of expensive peripherals, such as printers or hard disk storage devices, by several computer stations.

Aids in Management

On the practical side, a computer lab is a management aid. It centralizes the location of the hardware and software in the school building. This facilitates the cataloging, maintenance, and protection of these resources. Often one person has the responsibility of maintaining the computers and the related materials. The lab also aids in scheduling the use of the equipment, since the staff always knows where the computers are on any given day and can sign up to use them on one central scheduling form. It is easier to monitor proper student use of the equipment in a lab setting.

Promotes Equal Access

A lab setting promotes equal access to computers for all students. Without a lab, students whose teacher does not take the initiative to arrange for classroom computers may not have access to computers. When a small number of computers are available in a classroom, it is difficult for every student to have an opportunity to work at a computer. With a lab, teachers can schedule computer time for all students.

Facilitates In-service Training

Finally, computer labs can contribute to the goal of computer literacy for the staff. The lab provides an ideal setting for both formal in-service workshops on how to use computers and for brief, informal presentations and practice sessions.

Lab Purpose and Equipment Needs

Computer labs can be designed for a variety of instructional purposes. They may be categorized as programming labs, general instruction labs, or dedicated labs. The lab's objective determines the hardware and software needed in the lab.

Programming

Programming labs are commonly found in secondary schools and colleges. They are used to teach programming languages and computer science. They require computers that support the languages being taught (Logo, BASIC, Pascal, assembly language, COBOL, etc.). Programming labs are often networked so that students can have easy access to utility routines and can share printers. It is convenient to have a hard disk for storing standard routines and for quickly accessing students' files. Individual disk drives are needed so the students can make copies of their programs on floppy disks. Dual disk drives are important for ease in copying diskettes. Printers are essential for making program listings and recording output. The only software needed might be programming languages and utilities.

General Instruction

General-instruction labs are designed to teach a variety of content areas, such as language arts, science, social studies, and mathematics. These labs require significant amounts of appropriate software. In fact, it is recommended that the software requirements be one of several determining factors in the selection of computer hardware.

General-instruction labs are also used for computer literacy instruction, especially in the elementary grades. They provide an ideal setting for teaching students keyboarding skills and procedures for operating computers.

These labs can consist of either individual computers or networked computers, and there are advantages and disadvantages to both arrangements. Color monitors may be a good choice especially for providing motivation help at the elementary level. However, monochrome monitors are better for word processing. Printers are extremely useful and may be necessary for some applications.

Dedicated Labs

Although dedicated labs are appropriate in many subject areas that involve substantial computer use, they are most commonly found in business education departments.

Business education labs are intended to simulate the technical equipment in an office environment and to provide experience in business computer applications. Work stations typically include a computer of the type used by small businesses and offices, two disk drives, and a printer. Monochrome monitors are preferred for their ability to clearly display eighty columns of text per line. Word processing, database programs, and spreadsheet packages are the most commonly taught software. Networking can permit the sharing of printers.

Planning For Decision Making

Planning for a computer lab should be one of the outcomes of a district technology plan. The district's goals for using technology in instruction should be determined before the lab is planned so that the use of the lab fits into the district plan. Also, the lab for each building should be planned as part of the total K-12 program. It is crucial to determine the objectives for the lab before planning the location, software, hardware, or staff.

Creating a Computer Lab Committee

Creating an effective computer lab involves making a vast number of decisions. One effective way to make these decisions is through a committee. A committee can draw upon the experience and expertise of a number of people, and the committee members will feel a greater sense of ownership and support for the project.

Planning The Budget

Budget considerations are part of the initial planning for a computer lab. Most computer budgets contain funds for the purchase of hardware and software; however, maintenance, personnel costs, communications, and supplies are frequently overlooked. Some computer lab items may be part of separate budgets, for example, the English department may purchase word processing software, and the utility costs may be part of the overall building budget. In preparing the computer lab budget, consider the possibility of raising funds through grants, business donations, or PTA events.

Planning For Evaluation

Evaluating the computer lab is an important step in validating the expenditure of funds and other district resources. It also provides vital information to aid in making changes and improvements to the lab. The data collected through the evaluation may demonstrate the need for additional labs in the future.

The evaluation of the use of the computer lab may be part of the overall evaluation of the use of computers in instruction, but it is separate from the evaluation of student learning outcomes.

The committee should begin by deciding what aspects of lab use will be evaluated. Then they should determine what data-collecting instruments are needed and compile the necessary forms.

Some examples of useful data include logs of lab use and software use, as well as staff and student surveys.

Planning For Implementation

The computer lab committee should decide on a general timeline for the creation of the lab. One person, usually the district or building computer coordinator, is given the responsibility for implementing the plan. This person determines the specific actions needed, the people involved in each, and the timeline for the entire project. Regular reports to the committee on the status of the lab and the problems encountered will help the committee remain involved with the project.

Computerizing The Classroom Teacher's Tasks
(Burn Your Computer Out - Not Yourself!)

R M. Crawford
Bowmanville Senior Public School
105 Queen Street
Bowmanville, Ontario L1C 1M7

The computer has been touted for a number of years now as being the answer to almost every problem. Even the teaching profession is trying to assimilate the state-of-the-art equipment and 'prepare students for the future'. However, the educators themselves have been overlooked and many of their tasks suffer from not having any programmes that link together to support most of their responsibilities. The "Teachers' Administration Programme" makes an attempt to remedy this oversight.

No matter what programme you try you will have to adapt your methods to suit what it has already supposed is the right way or at least a better way. If you refuse to create for yourself what you need then you must shop around until you find the software that allows you to operate as close to the style that you are used to. The "Teachers' Administration Programme" (TAP) was created by myself and is written in Basic to accommodate as many styles as possible.

TAP is written as a turnkey system which means it runs from a main menu and you may move from area to area without having to shut down and manually load another programme for the next task you want to complete. There are presently eight choices at the Main Menu level:

1) Build and Modify Class Lists
2) Build and Modify Student Data
3) View Data and Print Forms
4) Enter Marks and List Averages
5) Enter and Examine Attendance
6) Create & Print Daily Lesson Plans
7) Setup the Default Information
8) Explanation of the Above Choices

Selection '8' gives the novice a quick overview of each module and some initial guidance. It is expected that familiarity with the modules will eventually allow you to ignore this choice.

SETUP THE DEFAULT INFORMATION:

To make the programmes run smoothly and professionally, it was necessary to create a module (choice 7) that would preset some information. This allows users with a variety of printers to get the best possible results. Choice '7' lets you enter your own name and that of your school, as well as its address. It presents an opportunity to enter the classes, their homeroom teachers, and homeroom numbers. An entry choice is also available for you to list the subjects that you teach. If you intend to track attendance unofficially you can also use the calendar setup routines so you may search for patterns in absence and lateness.

BUILD AND MODIFY LISTS:

Once the default information (choice '7') is entered, it is only logical to start entering the names (choice '1') of the students that you teach. This list of names is usable in many tasks that you perform through the year. Class lists and mark lists can be printed at any time and be as up-to-date as your last entries. If students want name tags for their texts and notebooks, the capability is there.

Changes are easily made and you can alphabetize the list upon your demand. At the conclusion of your entries for a class, there is the opportunity to get your first general list right away and check to make sure that no errors have occurred during the process.

BUILD AND MODIFY DATA:

The power of manipulating your information lies in this module (choice '2'). If you have a school office that handles most of the categorizing for you, this section may be redundant. If you decide to make use of it, you can create a file that stores most of the information about each student. With this section you can produce bus lists, age breakdown lists, medication lists, locker lists, mailing labels, and address envelopes.

Of course you don't have to fill in all of the areas. It's perfectly okay to keep a minimum of information and do only what it will provide.

VIEW DATA AND PRINT FORMS:

Completing your tasks is what choice '3' does best. When you are going to call this section, the bulk of your data entry (choice '2') should be done. The previous two choices have simply been entry and modification areas. This area prints it in the fashion you choose. Most of the impressive paper-work you show others will come from here.

Besides reading the data files and printing all the information about your classes, this module has some extra perks with it. Note-size forms can be produced with your name on each sheet. A grid pattern can be produced on a ditto and be used to print grids on all of your dittos. When you see the result, you'll know what to do with it. General and Subject lists are produced as well as standard forms for collecting information and trip permission.

ENTER MARKS AND LIST AVERAGES:

This is where the real work (choice '4') is for all of us and it's a pretty standard offering when others think of helping teachers with their work. However, this module doesn't just collect marks. You keep track of your tests, what they were out of, the date they were given, and get an updated average for each student that wrote it. Absence is accounted for and it is possible to allow rewrites and update the original mark.

You won't get variance figures or standard deviations, however there is a special routine that allows you to adjust the actual class average and redo every student's mark. You may never use it, but it's nice to have at times. Of course you may break your subject down into weighted sections and average them together to obtain a true mark. The possibilities are many and its flexability offers you as much ease as any programme can.

ENTER AND EXAMINE ATTENDANCE:

Choice '5' is not official and it isn't designed to be. But if you want to check for patterns in any student's habits, this programme will catch them. It separates their lates, absence, and excused absence into week days as well as school timetable days.

Entry is easy and quick. All student data can be entered at once and the computer will search the files and update them accordingly.

CREATE & PRINT DAILY LESSON PLANS:

This section (choice '6') draws from files you have created with it ahead of time and places the proper lesson on the sheet for the day. Headers are placed and the programme advances to the next lesson in your plan series.

If interruptions through the day prevented a lesson from being taught, you can have it listed again on the next plan sheet. It does limit you to five lines of 50 characters for each lesson, so don't plan on letting it be your full effort.

This module encourages you to set up a series of lessons that will run for the entire year and can be reused in future years and other schools. I suggest that you plan on paper (or with a word-processor) first and transfer your effort to the files.

TECHNICAL ASPECTS:

This programme is written in MBasic, version 5.2, for a Kaypro 4 computer. It uses character string ... CHR$( ) ... codes to control all of its functions. I have separated the screen and cursor control codes into subroutines so that most changes can be made in one place.

It shouldn't be too difficult to convert the screen to 40 or 64 column sizes as they have been kept simple to avoid clutter. If you don't have a Basic similar to that mentioned you may find that some commands are not available (eg. function calls ... IF - THEN - ELSE ...). On the other hand you may be working with a structured language that has much more power (Pascal, Logo, Modula-2).

The source code is available and commented if you would rather work from that, however it does cost me to print it and a charge to cover this will apply.

OVERVIEW AND PHILOSOPHY:

Computers and operating systems enjoy success not because of their superiority, but because of the freely distributed information about it so that support can come from as many areas as possible. CP/M, one of the most popular 8-bit operating systems, is an example of being raised to a level of a de facto standard for 8-bit systems because of the public domain programmes that exist for it. IBM's Personal Computer is not good because of a better than average design, but because they encouraged the creativity of others to support it.

It seems to me that everyone who writes a programme that is decent tries to break into the software business by selling it. Perhaps they have visions of leaving a boring job behind and making a quick financial killing in the high-tech business. This programme is free to all who are interested in using it. It works as described and when faults are found (parts of it have been used for over a year) they are corrected. I am not trying to set a standard for teachers, but I do encourage others to offer their personal programmes to the profession so that we can build a good set of administrative programmes that all can use.

This programme is not to be sold by anyone, however I do wish to emphasize two conditions that I would like to have met by anyone who uses it:

a) please distribute it as freely as it was distributed to you.
b) forward all new modules you create for your own use to myself so I may be the contact person to keep everyone up-to-date with the effort that is being expended on it.

No programme can be described as complete, but this one makes a claim that many others don't. It is free for the asking and I encourage you to make copies of it and contribute new modules to it so that we all may benefit as it grows.

COMPUTER STUDIES

Graduate Microcomputer Education: Another Perspective

Edward L. Anderson, Ph.D.
University of Wisconsin-Oshkosh
College of Education and Human Services
Oshkosh, WI 54901

The rapid emergence of microcomputers in educational settings has resulted in the need for new skills and knowledge by educators at many different levels. Higher education institutions were generally not prepared to meet this need and frequently, programs come into existence with little planning and little thought as to the needs of various educational populations. This paper will present another perspective in microcomputing education - a perspective for preparing individuals having different career goals while allowing for individual differences.

The initial purpose of the project was to develop a 15-21 credit area of emphasis in microcomputer education which would become an alternative area of study within the Masters of Science Degree in Curriculum and Supervision. The decision to attempt this development within an existing graduate program was based on the assumption that a totally new program would involve a difficult and lengthy process through the bureaucracy of the University System and would considerably delay any attempt at implementation. Additionally, some of the existing microcomputer courses were thought to be appropriate in the proposal program.

The project consisted of the following five independent, yet related, activities which were to provide background information which could be used in making program decisions.

Activity 1.

A mail survey was conducted of 417 graduate students who had completed a one-credit course entitled "The Microcomputer as an Instructional Tool". From a listing of thirteen topics, respondents were asked to indicate not only interest in a given topic, but to give first, second, and third choices which was intended to give direction to the writer as to which courses would need to be developed and in what sequence. A sixty-four percent response was received (N=260). A second set of questions was designed to determine interest in additional coursework and/or a graduate program. One hundred eighty-one (70%) indicated an interest in additional coursework and forty-three (16%) indicated an interest in a program leading to an advanced degree.

Activity 2.

Letters were written to the Association for Educational Data Systems and the Association for Educational Communication and Technology requesting their assistance in identifying graduate programs in microcomputer education. Neither organization responded.

Activity 3.

From a search of the literature (journals, periodicals, brochures received) twenty-three institutions were identified as having graduate programs in microcomputer education (Summer-Fall 1983). Letters were sent to these institutions requesting programs descriptions, brochures, etc. Eighteen responses were received. Analysis of the programs was extremely difficult due to the wide variety of course titles and opportunities for electives within a program. Thirteen institutions had graduate programs ranging from thirty to forty-eight semester hours. Three institutions reported a certification only program with fifteen to eighteen hours required. Two institutions reported no program. Most programs placed a heavy emphasis on programming languages with some being one-third to one-half the program.

Activity 4.

Conferences were held with representatives of the Wisconsin Department of Public Instruction who are involved in microcomputer education. Their suggestions and input were most helpful and several were directly incorporated into the proposed program.

Activity 5.

Fellow faculty from the College of Education and Human Services and the College of Letters and Science were consulted to gain their input into the development of a graduate emphasis in microcomputer education. Faculty at several other higher education institutions were also consulted. In all cases, their input and/or reactions to the proposed program were most helpful.

Upon completion of the analysis of information gathered from the above activities, the following program was proposed. It builds upon the four core courses (12 credits) in the existing Master of Science-Curriculum and Instruction program; recognizes the needs of both preservice and inservice educators; and more importantly, recognizes the need to prepare individuals having different career goals.

All students seeking the advanced degree (36 credits) would be required to complete the four core courses in the program. The program attempts to meet individual needs by providing wide latitude in student/advisor program planning consistent with the career goals and needs of the clientele served by the program. Three possible career goals related to microcomputer education were identified - (1) School District Coordinator of Microcomputer Education, (2) Teacher of Secondary Computer Science Courses, and (3) Classroom Teacher.

The following proposed programs should serve the needs of individuals having the above career goals and who also wish to complete and advanced degree.

Group One--Core Courses Required of all Graduate Degree Candidates:

17-710	**Foundations for Curriculum Planning** Concepts of curriculum as they are affected by educational ideologies, history, and culture, the nature of the learner and by current trends and issues.
17-720	**Supervision of Instruction** Introduction to the foundations, techniques, roles, and needed skills for effective supervision in a school setting. Views supervisor as leader for instructional improvement and as innovator in curriculum change.
17-732	**Multicultural Education for Leaders** The study of cultural diversity in American society as it relates to

the educational leader who carries out professional tasks in a wide variety of settings.

12-770 **Foundations of Educational Research**
Introduction to the concepts, tools, and procedures which are essential for planning and conducting research in education and related fields. Emphasis is given to the interpretation and analysis of research literature from the behavioral and social sciences.

Group Two--Courses Required for the School District Coordinator of Microcomputer Education

17-715 **Microcomputer as an Instructional Tool 1 cr.**
An introduction to the use of the microcomputer in education. The major emphasis is on computer awareness and developing student confidence in using the microcomputer as a teaching/learning tool.

17-607 **Evaluation of Microcomputer Software 2 cr.**
The use of selected evaluation instruments to evaluate microcomputer software/courseware and materials used in computer education courses.

17-609 **Instructional Applications of Microcomputers 3 cr.**
Examines the various uses of microcomputer materials in instructional setting.

17-716 **Fundamentals of Courseware Development 3 cr.**
Principles and procedures for developing interesting effective and efficient computer instructional materials.

17-XXX **Information Systems and Technology 3 cr.**
Examines the interrelationship and use of various information technologies in instruction. Computer networks, cable and satellite television, and video disk technologies will be explored.

17-XXX **Administrative Uses of Microcomputers 3 cr.**
Examines the various uses of microcomputers for administrative tasks. Word processing, spread sheets, filing systems, etc.

14-586 **LOGO for Teachers 3 cr.**
Examines the philosophy of LOGO and its implementation in elementary and secondary classrooms.

34-706 **BASIC for Teachers 3 cr.**
Introduces the student to the development of interactive computer programs using the programming language BASIC.

Group Three--Certification as a Computer Science Teacher

34-7BA	**Foundations of Computer Programming**	**2 cr.**
34-7PA	**PASCAL for Teachers**	**2 cr.**
67-7NA	**Numerical Analysis for Teachers**	**3 cr.**
34-7AS	**Microcomputer Assembly and Machine Language**	**3 cr.**
34-7DS	**Microcomputer Data Structures and File Processing**	**3 cr.**
17-607	**Evaluation of Microcomputer Software**	**2 cr.**

Group Four--Electives

17-7ME **Computer Education Methods**

	Practicum	**2 cr.**
17-606	**Developing CAI Instruction Materials Using PILOT Language**	**3 cr.**
17-608	**Microcomputer Authoring Systems for CAI**	**3 cr.**

The Master of Science-Curriculum and Supervision was designed as a leadership degree. The combination of core courses and the appropriate courses in an area of emphasis provides both leadership skills and content knowledge. This program of studies provides a broad base of computer knowledge and should allow an individual to function well as a leader in microcomputer education within a school district.

The program leading to certification as a computer science teacher, Group Three, was developed with the Departments of Mathematics and Computer Science and can be a stand alone program! This program provides greater depth in those areas which are essential in computer science instruction. For those seeking the advanced degree, the twelve credits of core courses would be required.

The program for a classroom teacher whose goal is to improve their microcomputer skills would be designed with an advisor and be based upon the knowledge and skill of the individual. Courses from any of Groups Two, Three, and Four would be appropriate. Should the individual desire to pursue an advanced degree, the twelve hours of core credits would also be required.

For individuals seeking the advanced degree, a culminating experience is required. This might be a thesis, seminar paper, or problem-solving project in conjunction with a culminating seminar.

The options are available. Students having different career goals can be accomodated. The possibility of changes in career goals can also be accomodated. As with all programs, modifications will be made based upon new developments in microcomputers and changing needs of those individuals within our service area.

NOTE: Courses identified in the above programs having complete numbers are currently being offered. Other courses are in the development stage.

What Are The Effects Of Computer Programming On Cognitive Skills?

Michael Land and Sandra Turner
National College of Education
2840 Sheridan Rd.
Evanston IL 60201

The authors examined research studies concerning the cognitive effects of learning computer programming to answer these questions: Does learning to program improve students' problem solving abilities? If so, does the programming language make a difference? How have researchers defined and measured problem solving? What are the implications of the research for teachers and researchers?

The purpose of this session is to review the research on the effects of computer programming on the cognitive skills of students. Proponents of teaching computer programming have argued that learning to program helps develop higher level cognitive skills. Feurzeig et al. (1981) have provided an extensive list of the cognitive outcomes expected from learning to program. These include logical thinking, understanding of concepts such as variable and function, experience with problem solving strategies, procedural thinking, ability to find and correct one's errors, and recognition that there is rarely one best way to do something.

Many researchers use the general term "problem solving skills" to describe these cognitive outcomes. However, one of the major problems associated with conducting research on the effect of learning to program on problem solving skills is how to define and measure problem solving. The literature indicates that there are as many operational definitions of problem solving as there are research studies.

Of the studies we reviewed, two focused specifically on the influence of programming on mathematical skills, seven on the influence of programming on more general problem solving skills, and two studies compared the cognitive outcomes of learning Logo and BASIC.

Among the first group of studies is the Edinburgh Logo Project (Howe et al., 1979), in which Logo programming was taught over a two-year period to 11 sixth-grade boys selected from the lowest level math class in a private school. During the second year, the Logo activities were especially designed to teach topics in their regular mathematics curriculum. Standardized pre- and posttests measuring mathematics achievement were given to the Logo group as well as to a control group selected from the next-to-lowest math group. The experimental group showed a little more improvement than the control group on basic math skills, but the reverse was true on a test of "math attainment." Thus, over a two-year period, the study provided mixed and inconsistent results on whether learning Logo affected math achievement.

Soloway et al. (1982) asked half of the freshmen and sophomores in a programming class to solve an algebraic word problem by writing a computer program and the other half to represent the solution by writing an algebraic equation. The group writing the program scored significantly higher. These results provided support for their hypothesis that students make fewer errors in interpreting and solving algebraic word problems when they use a more active, procedural approach such as that required by programming.

In a second study, they compared the results of reading and explaining an equation embedded in a computer program with reading and explaining an algebraic equation that stood alone. The number of students who explained the computer embedded equation correctly but not the algebraic equation was significantly greater than the number of students who answered the algebraic problem but missed the computer problem. Again, the programming environment facilitated the students' understanding of algebraic problems.

In one of the first studies to investigate the effect of learning programming on more general problem solving skills, Foster (1972) administered the Problem Solving Abilities Test (PSAT) to 68 eighth graders. He used intact classes. Group 4 studied both programming and flowcharting; Group 3, programming only; Group 2, flowcharting only; Group 1, neither. The PSAT measured 9 behaviors associated with problem solving, such as proposing an hypothesis, identifying a pattern, specifying missing or irrelevant information. Foster found that Groups 3 and 4 did better than Group 2 and all three did better than the control group; however, an analysis of variance showed no significant main effect for the treatment.

In another study with a stronger research design, Statz (1973) gave pre- and posttests to 16 fourth graders who participated in Logo sessions and to a similar group who remained in the regular classroom. The problem solving test consisted of four parts: the twenty-questions game, the Tower of Hanoi puzzle, a task to generate permutations, and a word classification task. Statz found that students who learned Logo performed significantly better than the control group on the permutation and word classification tasks.

Seidman (1981) investigated the effect of learning programming on logical reasoning skills. Forty-one fifth graders were randomly placed either in a group taught Logo or in a group that received no special instruction. After the treatment, both groups were given a series of tests measuring their understanding of conditional statements (statements which use "if," "if...then," "only if," and "if and only if"). No significant differences were found in the two groups.

Gorman's (1982) study of the Lamplighter Project in Texas was similar to that of Seidman in that both were concerned with the effect of learning Logo on logical reasoning skills. In the Lamplighter study, Gorman used a "rule-learning" test. Children were shown a series of shapes in different colors and sizes and then were asked what features satisfy the rule of the experimenter. For example, pictures of red circles and a variety of nonred shapes fit the conditional rule "if red, then circle; if not red, then any shape." Third graders were randomly assigned to one of three classes. Two of the classes studied Logo one-half hour per week and the other class one hour per week. At the end of the school year, the students who studied Logo one hour per week performed significantly better than the other two groups on the task of learning the conditional rule.

In a carefully designed study of 40 fifth graders, Rose (1983) also found that programming instruction improved logic skills (as measured by the Cornell Critical Thinking Test), especially in the areas of hypothesis testing and determining the relevance of data. However, subject-oriented problem solving skills (as measured by the Developing Cognitive Abilities Test) were not improved with 26 hours of programming instruction.

In widely reported research at Bank Street College of Education, Pea & Kurland (1983) compared the planning abilities of 16 children, ages 8-9 and 11-12, who had Logo experience with a control group of 16 children of the same age who had not studied Logo. Students in both groups were asked to use a map of the classroom to devise an efficient plan for doing familiar chores, such as cleaning the blackboard and moving furniture. There were no significant differences between the groups on the efficiency of the plans, the quality of the revisions, or the types of decisions during the planning process.

Clements (1984) studied the effects of learning Logo on the cognitive abilities of first graders. Eighteen first graders were randomly assigned to either a Logo programming group or a CAI control group. Pre- and posttests were given to determine the children's level of reflectivity (thinking before answering), divergent thinking (creativity in drawing), vocabulary, metacognition (thinking about their own thinking), logical thinking (measured by Piagetian tasks involving left-right orientation, verbal and numeric memory, and drawing a design), and ability to describe directions using a simple street map. Although the two groups were equivalent in school success before the treatment, after 12 weeks the Logo group significantly outperformed the CAI group on reflectivity, divergent thinking, metacognition, and describing directions but not on the other measures. Clements concluded that although Logo programming may not affect the cognitive development of first graders when compared to CAI, it may affect the way in which children use the cognitive abilities they have.

In summary, the research to date provides inconclusive and apparently contradictory results on the cognitive benefits of learning to program. The studies are difficult to compare because their operational definitions of problem solving are so different. Even in the two studies (Seidman, 1981; Gorman, 1982) that investigated the effect of learning Logo on students' understanding of "if...then" statements, the assessment instruments were very different and the results were contradictory.

Tetenbaum & Mulkeen (1984) have pointed out that training in problem solving skills does not generally transfer from one task, such as programming, to success on another task, such as interpreting conditional statements or solving the Tower of Hanoi puzzle.

Simon (1980) has noted that transfer is enhanced when a) some of the skills used in the second task are identical to those learned in performing the first task, and b) when the learner is made explicitly aware of the skills involved, apart from the specific content of the task.

Thus, the implications for programming teachers is to make more explicit the links between problem solving in the context of programming and problem solving in other contexts. The relevant questions for researchers are a) what kinds of cognitive skills should be measured and how, and b) what are the salient features of the programming classroom environment that facilitate the transfer of problem solving skills to other kinds of tasks?

References

Clements, D. H. 1984. LOGO programming: Can it change how children think? **Electronic Learning, 28**, 74-76.

Foster, T. E. 1972. The effect of computer programming experiences on student problem solving behaviors in eighth grade mathematics. University of Wisconsin, **DAI, 33**, 4239A.

Feurzeig, W., Horwitz, P., & Nickerson, R. S. 1981. Microcomputers in education (Report No. 4798). Prepared for: Department of Health, Education, and Welfare; National Institute of Education, and Ministry for the Development of Human Intelligence, Republic of Venezuela. Cambridge, MA: Bolt, Beranek and Newman.

Gorman, H. Jr. August 1982. The Lamplighter Project. **BYTE**, 331-332.

Howe, J., O'Shea, T. & Plane, F. 1979. Teaching mathematics through Logo programming: An evaluation study. Department of Artificial Intelligence, University of Edinburgh, Forest Hill, Edinburgh. DAI Research Paper No. 15.

Milojkovic, J. D. 1984. Children learning computer programming: Cognitive and motivational consequences. Stanford University, **DAI, 45**, 385.

Pea, R. D. & Kurland, D. M. 1983. **Logo programming and the development of planning skills.** New York: Center for Children and Technology, Bank Street College, Tech. Rep. No. 16.

Rood, M. K. 1983. Locus of control, self-esteem, and the acquisition of problem solving skills through the instruction of Logo and BASIC computer programming languages in gifted children. The University of Connecticut, **DAI, 45**, 469.

Rose, N. S. 1983. The effects of learning computer programming on the general problem solving abilities of fifth grade students. North Texas State University, **44**, 2354.

Seidman, R. H. 1980. The effects of learning the Logo computer programming language on logical reasoning in school children. Syracuse University, **DAI, 41**, 2249.

Simon, H. 1980. Problem solving and education. In D. T. Tuma & F. Reif (Eds.), **Problem solving and education: Issues in Teaching and research.** Hillsdale, NJ: Lawrence Erlbaum Associates, 81-96.

Soloway, E., Lochhead, J., & Clement, J. 1982. Does computer programming enhance problem solving ability? Some positive evidence on algebra word problems. In R. Seidel, R. Anderson, & B. Hunder (Eds.), **Computer literacy: Issues and directions for 1985.** Academic Press: New York.

Statz, J. 1973. The development of computer programming concepts and problem solving abilities among ten-year-olds learning Logo. Syracuse University, Unpublished doctoral dissertation.

Tetenbaum, T. J. & Mulkeen, T. A. November 1984. Logo and the teaching of problem solving: A call for a moratorium. **Educational Technology**, 16-19.

QNX And The ICON: Security That Works

David Trueman
M.M. Robinson H.S.
2425 Upper Middle Road
Burlington, Ontario L7P 3N9

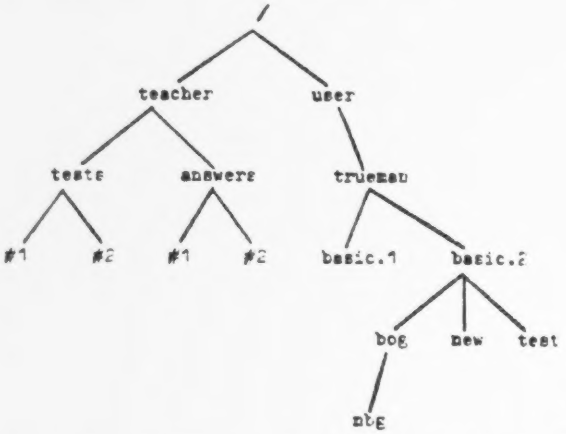
The QNX operating system as implemented on the ICON educational microcomputer affords the user several security features of considerable benefit in the educational environment. The structure of files and their security is outlined as well as the salient features of the LOGIN and PASSWORD system are discussed in connection with the "tree" structure of the file handling of QNX.

QNX, developed in Ottawa, Canada, by Quantum Software Systems, is a sophisticated, multi-tasking, multi-user operating system. Like most such systems, QNX relies on the storage of information in files. Each file must be assigned a filename of up to 16 characters in length with no imbedded spaces, although the "." and "_" are permitted as well as the numerals 0 to 9. The files are built hierachically using the usual "tree" structure with the "root" at the top.

Although QNX regards every file in its control as a randomly accessible series of bytes, some files, called directories, are used by the system to locate files through the one-to-one correspondence with their filenames in these directories.

Each user may create his own sub-directories to contain his files. An advanced computer science student, for example, working in both Waterloo BASIC and Waterloo Pascal, could have two sub-directories under his own directory, called "basic.1" and "pascal.1". The inclusion of the ".1" in the directory's filename could allow the student to create further sub-directories such as "basic.2", "basic.3", etc. Directories and files may have the same filename if they are not under the same directory.

The path QNX takes from the "root" to the specified directory or file is called the "pathname".



In the diagram above a pathname could be /user/trueman/basic.1, where the path begins at the "root" of the "tree" and then goes to the directory "user" where it finds the directory "trueman" then to "basic.1"

Like other multi-user systems, QNX uses a system of identification and passwords. These are assigned to the users by a system administrator and, with careful planning, the security of the system can easily be kept intact. When the system is powered up the ICON logo appears on the terminal along with the date and time. The system then prompts the user with

```
LOGIN:
```

to which the user is expected to respond with his identification previously assigned by the system administrator. The next prompt ...

```
PASSWORD:
```

requires the user to enter his pre-assigned password code.

The use of identification and passwords in the educational environment has distinct advantages for both the teacher and the student. It protects the teacher's work area where she may wish to store test answers, teaching examples, homework solutions, etc. QNX's LOGIN and PASSWORD system also protects the student's workspace from unwanted tampering by other students and builds a sense of ownership and pride that is so often missing in today's computer science classes where plagarism is endemic.

Pathnames can be assigned to PASSWORDS so that individual students can be directed to specific places on the disk or to their own work areas within the operating environment of QNX as soon as they login with their password. Each student in a typing class, for example, could be directed to the ICON editor, a powerful word-processing package employing the "window" technology so popular on the major word-processing machines today. A class of senior computer science students could be directed, when they sign in, to BASIC (structured Waterloo BASIC), to Pascal (Waterloo Pascal), to LOGO, or to the ICON editor to program in the C language, since all these languages are resident in the QNX utilities.

If the user has not been directed to any specific utility or program upon login, he will encounter the prompt character ...

```
%
```

This indicates that the ICON is now ready to accept the user's next command.

Should he not know where he is in the "tree", he may enter ...

```
% wmi
```

the classic "Where am I?", to which QNX will respond with the pathname of the user such as

```
/user/trueman
```

An attribute is defined as a field attached to every file which describes the access modes available to the owner of the file. Attributes consist of read, write, append, execute, and modify. Permission is a second field attached to each file which describes the access modes of those who are not the owners of the file. The range of permissions (granted by the owner) is the same as that of the attributes. Directories also have attribute and permission fields attached to them. Attributes which can be assigned to a directory are read, create, modify, and block. The assignable permissions for directories are the same as the attributes.

By using the attributes and permissions fields attached to each directory and file he creates, a user can protect his work in two ways. He may put a per- mission block on a directory which contains his sensitive material. Once this is done, anyone trying to get a list of his files will be prevented from doing so, although he himself will have full access. The second way of preventing access to sensitive files is to place a block permission on a directory. This prevents anyone but the owner from doing anything below that directory. Even if someone knew the name of a file, that person could not "reach through" a block to access it in any way.

Attributes and permissions on files are changed through a QNX utility called CHATTR. To use this utility you must be on the "branch" above the file or directory whose attributes or permissions you wish to alter. For example, if you wish to change the attributes on the file "basic.2" whose pathname is ...

```
/user/trueman/basic.2
```

you must first of all own the file and be one level above the file at

```
/user/trueman
```

in order to affect the change. In the latest version of QNX, permissions may be granted to specified groups (of students, for example, or classes) or simply to other individuals.
The format for using the CHATTR utility is ...

```
chattr <path> <name> pg=<options>
```

The options are read, write, append, execute and modify. It should be noted that unless you never wish to access your file again, you should not add a block to the attributes of your files. The "m" for modify should never be removed if the file is to be altered in any way. Of course, if the files are permanent, and you never wish to change them, the removal of the "modify" attribute from your files will add one more level of security.
The same format is used to give (or remove) permissions from files. The CHATTR utility is also used to change the attributes and permissions of directories using the same format as above.
To determine what files and directories are "below" a specific level in the hierarchy, in the latest version of QNX, the command...

```
% files
```

will bring to the screen, not only the name of the files and directories, but also the date and time each was created as well as the size of the file. The listing on the screen would look something like...

```
BLK Date      Time   Name
 2  1-Dec-84 2:23pm test
 3  2-Dec-84 2:25pm bog
 1  8-Dec-84 1:15pm /bog/nbg
```

The first two entries in the example are directly below the level for which the command "files" was given. The last entry in the example is an illustration of a file stored "deeper" in the "tree". Note that the full pathname is given and it is stored below the directory "bog". If you would like a more verbose listing of the files stored beneath the branch you are on, the command...

```
% files +v
```

where the +v stands for "more verbose", will give you, in addition to the above information, the heading "X", which stands for the number of "extents" in a file. If, for example a file is spread over the disk, the number listed below the "X" heading indicates the the number of locations in which it is stored. The heading, "Loc", gives in hex the starting location of the file. The heading, "Grp", gives the group number to which the file belongs. The heading, "Own", states the number of the owner within the group who created or "owns" the file. "Attr", lists the attributes assigned to the file, and beneath the heading "G-Perm-O", is listed the permissions assigned to the group and the permissions assigned to others (outside the group).
Before changing the attributes or permissions fields of a file, the user should always check the current status of these fields by using the command "files +v".
The reader should understand that the system administrator can override all permissions on all directories and files and can, therefore, can read, write, append, execute, and modify all users' files. She is the only one who can do this. All other users of the system must abide by the permissions and attributes built into the files.
In the most recent version of the QNX operating system all users may go to any branch of the "tree" by using the "change directory" command. If for example a user is at the "basic.2" branch and he

knows there is a "tests" file under the "teacher" directory. To reach the "tests" file he simply enters ...

```
% cd /teacher/tests
```

and, without any security, all the intruder must do to list all the tests in the file is to type ...

```
% ls
```

which calls up a complete listing of all the tests stored in that file.
The owner of the file need only to invoke the "block" through the "CHATTR" utility ...

```
% chattr /teacher/tests  po=+b
```

and when our interloper now tries to access the files he is met with the terse ...

```
:permission denied
```

Even though he may know the name of a file below the blocked directory, he cannot access it.
The security of the system is excellent but is very much dependent on students and teachers accepting the responsiblity for the security of their own files. I strongly recommend, therefore, that wherever ICONS are being used and running QNX, that all teachers using the system become very familiar with the security available to them. Once the system is understood, and its benefits and operation explained to the students, they will embrace it eagerly and breathe a sigh of honest relief for the privacy it provides.

In these few paragraphs I have attempted to give the reader a sense of the security offered by the QNX operating system used on the ICON. It is at best a very broad brush and I would direct the interested reader to the following sources...

BYTE
Vol 2, Aug 83
This issue was devoted to articles on UNIX (a registered trademark of Bell Labs) and is of interest since QNX works in much the same way.

QNX for the ICON
Sept 84
This manual is available through Burroughs Canada, runs to nearly 200 pages, and gives a complete description of the QNX operating system as it is implemented on the ICON.

Training Teachers Of Secondary Computer Science - A Methods Course

Henry S. Kepner, Jr.
University of Wisconsin-Milwaukee
Dept. of Curriculum and Instruction
Milwaukee, Wisconsin 53201

As computer science is becoming a study of great interest in the secondary school, teachers must be adequately trained to teach computer science and related topics to students. In addition to training in computer science topics, prospective teachers in this field must investigate methodology and curricular questions in this discipline. This presentation includes topics, materials, and issues in the training of computer science teachers.

With the new emphasis and interest in the teaching of computer science in secondary schools, the profession must urgently address the preparation of teachers in this field. This paper will identify appropriate curriculum topics for the teaching methods and curriculum course for prospective teachers of computer science.

At the present time, students completing a major or minor in computer science are frequently ill-prepared to move directly into the teaching of computer science in secondary schools. In particular, they have had little preparation in deciding what is important and appropriate to teach in the secondary school. In addition, they have no preparation in effective ways of presenting these topics to students with varying levels of maturation and sophistication in such topics. Additionally, prospective teachers have little training in dealing with the equipment, laboratory and administrative responsibilities assigned to them.

First, prospective teachers must review computer science as an emerging discipline with many varied areas of study. Students who are just finishing an undergraduate curriculum seldom have a thorough awareness or understanding of how the varied aspects of the field are related. This is particularly true of students who are completing an academic minor in computer science. A computer science methods course should include an overview of the several areas of computer science.

The field of computer science is so new that there are not adequate models of curriculum at the secondary level. In contrast to existing fields such as mathematics, physics, and so on, there are no standard texts and accepted sequences of study for the beginning teacher to use. A methods course must identify some common practices and prepare the student to develop curriculum from the start of a teaching career in computer science. This ability to build and modify a curriculum will likely be used for many years to come as topics and technology continue to develop and change at a rapid pace.

In order to respond to one of the most controversial questions in schools today, the prospective teacher must study the various definitions and views of computer literacy and computer awareness. In most teaching settings, this will be a primary area that the new computer science teacher will be asked to address. As many states try to define and mandate computer literacy requirements, the prospective teacher will be assigned to present ideas and to develop curriculum for this topic as well as teach computer science topics to advanced students. Without adequate investigation, the typical computer science major will have an academic view of computers which will not be beneficial to the entire student population.

One of the most detrimental views of computer science in secondary schools today is that it is centered on programming the computer. There must be a clear presentation that problem solving must be the focus of computer work in the schools. With the ease of initial programming in BASIC, the computer science teacher has a major task to overcome student tendencies to start writing lines of code before clearly organizing a plan to solve the problem at hand. In addition, the prospective teacher must realize that many problems can be assigned in parts, not to be carried to completion. Some student assignments can require only a plan - pseudo-code, flow charting or narrative. Others may focus on the maintenance of an existing program, debugging a program written from a plan, or testing the program under various sets of data. Currently, most teachers and nearly all students see line-by-line coding as the end all in computing. The focus on planning and algorithm construction is critical.

In writing programs to solve problems, the student must be introduced to the latest techniques in design. Currently, work in top-down design and modularization are the most important factors. Others styles should be available as the tasks become more complex. The emphasis in construction should center on good documentation and writing style. It is clear that programming instruction is just another form of writing. We should not be afraid to learn from the literature on composition to help students become better writers -- whether the task is coding, pseudo-code, documentation or prose.

As a follow-up to the focus on problem solving, algorithms, and good writing style, prospective teachers should be thoroughly aware of the content and direction of the Advanced Placement program in computer science. While this program is not dealing with material for all students, the advanced placement program is a good barometer of the developing freshman expectations for computer science students at the college level. Consequentially, the high school curriculum should reflect a knowledge of this direction. On the other hand, other directions should be noted, too. One of the largest employers of computer qualified students will continue to be data processing and other business fields. The expectations in these areas will not be identical to those in departments of computer science. Since most students taking computer science courses in high school will not be going into a computer science major, the alternative expectations must be considered.

As one part of a total program in computer science, secondary students must learn to control the computer at several different levels. With the wealth of computer languages available, students should have some competence in at least two high-level languages during their school career. Today, the most likely options include Logo, BASIC and Pascal. It is important to note that these languages are constantly changing as new implementations are made. In many ways, each is adopting strengths of the others that were not included in the original implementations. Newer forms of BASIC include much of the structure developed in Pascal. New versions of Pascal have greatly simplified the input/output capabilities of earlier versions. While purists decry these modifications, the final result is that more users will be able to use the language most suited for their tasks. Advanced students should have some exposure to work in assembly language. The purpose of this exposure is to help students see the complexity required to have maximum control over the computer. Mastery at this level is not recommended. Advanced study in assembly language should be on a need-to-know basis.

A new level of language study is just becoming available to the secondary school program. The use of menu-driven software is becoming a major tool for students at many different levels of technical knowledge. Through word processing software, instructional programs, or data-bases, students are learning to use the computer as a tool by calling the special functions that will do the task required. Use of such software is definitely a problem solving skill for the student involved. The new software allows a student to plan and implement a solution to a problem thorugh use of the major functions of the software without the need to write lines of code. The student is often learning all the skills needed to work with the construction, transfer and maintenance of files without writing lines of technical code that apply to a single computer format. Again, the stress is on using powerful computer subprograms to carry out the task designed by the student. Learning to program at this level should be accomplished by all students in their study of the computer. The software chosen should be based on the most appropriate applications for the student.

Advanced study in the use of data and data manipulation leads the student into basic principles in data structures and computer architecture. Here, the emphasis is first on the appropriate way to conceive of the data in an organization for dealing with it most easily in the outside world. Next, the focus must shift to find ways to store the data in a computer to meet the application demands.

A major area of social issues raised by computer technology is often ignored in a computer science or computer engineering program. Prospective teachers must be trained to deal with social and ethical issues of computer technology in the secondary school curriculum.

First, there is the need to present teachers with the basic information on social implications of the computers in our society. This can be done through readings and specific identification of uses and abuses as the technology and related techniques are presented.

Second, there is the need to train teachers to help students deal with these issues. It is clear that we do not have the answers to many of the technology issues of today. Through techniques of values clarification, teachers can help students learn how to identify the alternatives and to value them in a discussion setting. In most areas - such as copyrights, access to information and the cost of information - there are no definitive answers, we must be careful not to preach a correct solution. Students must be taught to deal with our democratic process and to work for change of unacceptable rules in our society. Some work in values clarification will be needed by those responsible for dealing with these topics in the secondary school. Now, few mathematics/computer teachers have this skill.

While the ACM recommendations on the training of computer science teachers chooses to ignore the administrative and general consultant nature of a computer science teacher, this author sees these areas as critical to meet school demands on the individual. As the qualified computer teacher in a district, the computer science teacher will be expected to administer the computer laboratory and all related equipment. Consequently, that person should have some training in the various tasks related to this job: writing budget proposals; planning for equipment purchase, maintenance and replacement; organizing and staffing computer laboratories; providing consultant services to the administration and staff; and evaluating equipment, curriculum and personnel. These numerous responsibilities do not all come from the basic training in a computer science sequence.

It is hoped that many of the listed functions will eventually be shared by others in the district, these will typically be assigned to the new computer science teacher in many school districts - especially small ones. Some preparation in administrative skills and in dealing with general uses of computers in the total school is needed.

At this time, the methods course, or additional coursework in computers in education or general curriculum development are the typical places to present minimal preparation for the school-wide demands placed on the incoming computer teacher.

The prospecitve computer science teacher should have some background in a variety of computer uses in the various disciplines. The typical computer science curriculum does not always show these varied applications in a way that the student can transfer them into applications that can be shown to secondary school pupils.

We must seek to be more successful in showing computer students how the discipline is used than we have been traditionally in mathematics at the high school level. In presenting the typical topics such as searching and sorting algorithms, the teacher must be sure to allow students to see how these are general algorithms that have many applications in a wide variety of applications throughout the range of curricular areas - and beyond. If we are not successful at this look at applications, the number of students who pursue computer coursework will quickly dry-up after the current public fascination with computers diminishes.

At some point in the college student's preparation, the prospective teacher should be required to serve as a computer consultant to a person who needs an application of the computer. Whether through writing a program or adapting existing software, the teacher should have the opportunity to build a piece of software to the specifications of a user. This skill is one of the most important to all computer scientists who will work to apply their tool to societal needs. Practice in communication at this level is particularly important to the prospective teacher. If not included in the computer science sequence, this communication task is another that must be attempted in the education part of the program.

A final area of discussion for the prospective computer science teacher is that of continued professional growth. The field of computer technology and applications is growing and changing so rapidly that one can not hope to keep up in all areas. In a few years, one can easily be out-of-date in many aspects of computer use and knowledge. Probably more in this area than any other discipline taught in the secondary school, the teacher has an obligation to be aware of changes and to be willing to continually modify the curriculum.

Unfortunately, the educational establishment is not geared to this fast change in discipline and the need for continual revision of content and materials. Consequentially, the teacher in this area should be taught to fight for extra resources, professional time, and conference participation. Proposals to the adminsitration are needed to give the teacher adequate time to keep up. Additonal summer curriculum writing time may be needed to revise the instructional program and student materials.

After considering the above areas where additional training is necessary for the teacher of computer science, it is recommended that at least one methods and curriculum course is needed beyond the basic training in computer science.

Unfortunately, this recommendation is far beyond the minimal requirements in most states and provinces where there are no requirements beyond teacher certification in some other field. It is clear that we should not expect a teacher trained in mathematics or some less realted field to have the range of skills cited above.

Note: A list of suggested readings in the areas cited in this article is available from the author. Responses and suggestions are welcome.

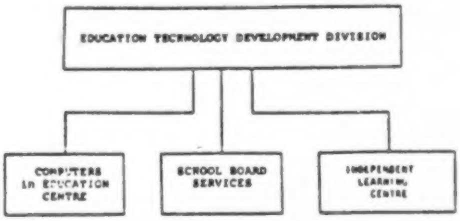
Computer Studies In Ontario Schools: Curriculum, Hardware And Software

Dave Didur, Co-ordinator of Computer Studies
Ontario Ministry of Education
24th Floor, Mowat Block, Queen's Park
Toronto, Ontario M7A 1L2

Students must be prepared to respond to the changes in the workplace, in technology, and in society caused by the computer revolution. This will not be an easy task unless educational institutions are prepared to do the same. The Ontario Ministry of Education has undertaken numerous initiatives to prepare for this state of change, including a reorganization of its own structure, the revision and updating of its Computer Studies curriculum guideline, and hardware and software initiatives to support new programs and to provide flexibility for future directions.

1. ORGANIZATION

In December, 1983 a new division within the Ministry of Education was created, the Education Technology Development Division.


EDUCATION TECHNOLOGY DEVELOPMENT DIVISION
COMPUTERS IN EDUCATION CENTRE
SCHOOL BOARD SERVICES
INDEPENDENT LEARNING CENTRE

1.1 Computers in Education Centre (CIEC)

- . responsible for the development and application of computer technology in Ontario education, particularly in the areas of hardware, software, instructional competency, and the Computer Studies guideline

1.2 School Board Services Unit

- . responsible for the provision of data processing services and computer-based management services to the school boards of Ontario, especially via the Educational Computing Network of Ontario (ECNO)

1.3 Independent Learning Centre (ILC)

- . responsible for facilitating the development and implementation of high technology delivery modes for curriculum programs

The co-ordinated activities of the Computers in Education Centre and the Independent Learning Centre, as they relate to the Computer Studies guideline, will be highlighted.

2. CO-ORDINATED ACTIVITIES

Three major activities related to the Computer Studies program are illustrated below:

2.1 Curriculum Guidelines

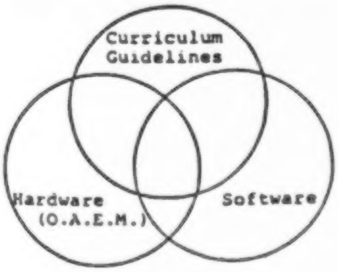
. a new Computer Studies guideline was released in December 1983 after three years of development

2.2 Hardware

. Ontario's functional requirements for an Ontario Approved Educational Microcomputer (O.A.E.M.) were satisfied with the production of the ICON, a state-of-the-art microcomputer approved for use in Ontario schools in April 1984.

2.3 Software

. a two-year project concerned with the production of exemplary lessonware ended in March, 1984, and a $25 million software initiative over the next three years was begun



3. COMPUTER STUDIES GUIDELINE

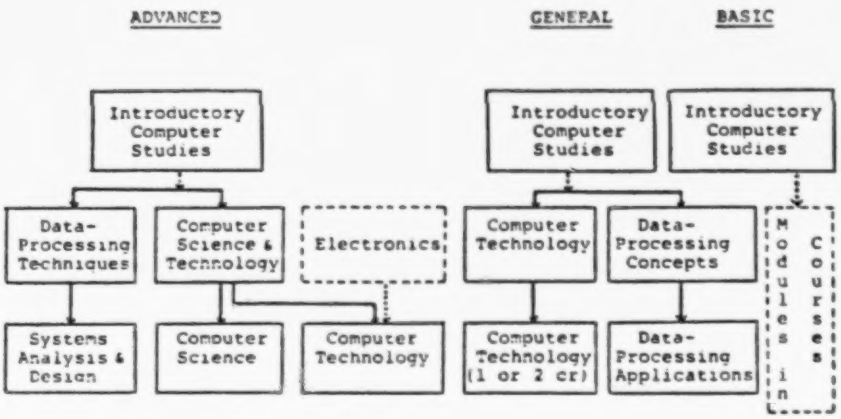
In December of 1983, the Ministry of Education released a new curriculum document entitled "Computer Studies, Intermediate and Senior Divisions, 1983." This one document superseded four previous guidelines:

- Computer Science, Senior Division, 1970;
- Data Processing RP.33, 1966;
- Elements of Computer Technology, Senior Division, 1970;
- Informatics, Intermediate and Senior Divisions, 1972.

The new guideline takes the first step at synthesizing what has traditionally been very distinct areas of study. Many of the courses integrate material from business studies, mathematics, and technology programs. No other field of study is likely to be in such a state of flux in the coming years as is Computer Studies, and it is expected that the philosophy and direction of this document will be further shaped by future change.

BEST COPY AVAILABLE

COMPUTER STUDIES PROGRAMS


ADVANCED
GENERAL
BASIC
Introductory Computer Studies
Introductory Computer Studies
Introductory Computer Studies
Data-Processing Techniques
Computer Science & Technology
Electronics
Computer Technology
Data-Processing Concepts
Modules in Courses
Systems Analysis & Design
Computer Science
Computer Technology
Computer Technology (1 or 2 cr)
Data-Processing Applications

Each block represents the authorization for a single-credit course, except where noted otherwise. The arrow-lines indicate recommended prerequisites.

3.1 The following points highlight some of the features of this document:

- 3.1.1 There are courses at three levels (Advanced, General, and Basic).
- 3.1.2 The recommended grade levels are 10, 11, and 12, but flexibility is permitted.
- 3.1.3 The Introductory courses are quite broad in scope, including material from each of the senior division courses (Data Processing, Computer Science, Computer Technology).
- 3.1.4 Specialization is possible in the senior division.
- 3.1.5 The Computer Science and Technology course (grade 11) is a recommended prerequisite for both grade 12 Computer Science and grade 12 Computer Technology (Advanced).
- 3.1.6 There are two different streams in Data Processing. The advanced courses emphasize programming and program development. The General level courses are applications courses in which the emphasis is on operating computers and using business software packages.
- 3.1.7 All programming work involves a structured programming approach, with top-down development, modular design, data-directed design, pseudocode, and good coding style.
- 3.1.8 Emphasis on communication skills is stressed in all courses.
- 3.1.9 For each of the twelve courses, the document gives objectives, recommended timelines, and teaching suggestions. Evaluation criteria for each level are also provided.

3.2 The following activities in the province are concerned with the development of resource materials for the document:

- 3.2.1 A co-operative project involving members of ALSBO (the Association of Large School Boards of Ontario) resulted in the development of complete courses of study (including lesson plans, assignments, background information, etc.) for two of the three Introductory Computer Studies courses. Work on a similar project for the gr. 12 Computer Technology courses began in 1984.
- 3.2.2 The Ministry of Education has funded the development of resource materials lists for all the senior division courses, and has developed a comprehensive teacher's guide for teaching the technology module in the gr. 11 Computer Science and Technology course (available in French and English).
- 3.2.3 The Educational Computing Organization of Ontario (ECOO) has developed a resource booklet for use in the Computers in the Classroom courses, but teachers of computer awareness or literacy programs may find some useful material in it.
- 3.2.4 The Ontario Educational Communications Authority (OECA), commonly referred to as TVO (TVOntario), has developed audiovisual support materials for the Introductory courses. Both the English series "Nibbles" and the French series "Puceron" are supported with student texts, and content is cross-referenced to the guideline.
- 3.2.5 The Independent Learning Centre has begun a project to develop seven remote distance courses for Computer Studies, including software to support various topics within these courses.

School boards are expected to have all grade 10 and 11 Computer Studies programs based on the new guideline by September, 1985, and all grade 12 courses based on the guideline by September, 1986.

BEST COPY AVAILABLE

4. HARDWARE

Rather than put the horse behind the cart by trying to adopt a presently existing commercial microcomputer as the provincial standard machine for use in Ontario Schools, the Ministry of Education issued "state-of-the-art" specifications outlined in Functional Requirements for Microcomputers For Educational Use in Ontario Schools in March, 1983. The ICON is the first system which meets these specifications. Developed by CEMCORP, a Canadian company specially formed to design systems and software specifically for educational use, the ICON is marketed and supported by Burroughs under a joint venture agreement signed in October 1984. Over 4,000 of these systems were installed in Ontario schools by the end of 1984.

This is a powerful system which will not only dramatically improve the quality of educational software for classroom instruction, but will also dramatically impact on the activities in the Computer Studies program. The machine specifications include:

- 16 bit Intel 80186 microprocessor
- 384K RAM
- built-in trackball
- sculptured bilingual keyboard
- adjustable screen, pedestal mount
- voice synthesizer
- Telidon NAPLPS graphics protocol
- QNX operating system, supporting multi-users and multi-tasking
- screen segmentation into windows
- a sophisticated user interface using graphic images, or "icons"
- networking arrangements using coaxial cable connected to a fileserver (hard disk) and ARCNET protocol
- optional standalone capability with a built-in double-sided, double-density floppy disk

A comprehensive program of grants is available to school boards for the purchase of O.A.E.M. hardware.

5. SOFTWARE

It has already been mentioned that the ICON will have the QNX (a UNIX-like) operating system, NAPLPS graphics protocol, and a sophisticated user interface using icons and pointers manipulated by a trackball. In addition, the machine will come with the programming languages Logo and C, as well as the following University of Waterloo structured languages: WATCOM BASIC and WATCOM Pascal, and (with 512K RAM) WATCOM FORTRAN, WATCOM COBOL, and WATCOM APL. These last five are all enhanced versions (V2.0) of the Waterloo languages. Graphics utilities and an iconic text editor also come with the system.

In addition, the Ministry of Education and the Ministry of Industry and Trade, using funds from Ontario's Board of Industrial Leadership (BILD), have embarked on a comprehensive software initiative to develop educational computer-based learning materials for any O.A.E.M. The following projects have applications in the Computer Studies program:

1. CASI...Computerized Accounting Simulation
   - could be used as an application in D.P.
   - presently exists for Commodore 8032s
   - ready for the ICON September '85
2. Logic Circuit...build & test digital logic circuits
   - uses in Introductory Computer Studies and C.S.& T.
   - presently available for Apple computer
   - ready for the ICON September '85
3. ALICE....Algorithmic Language Interactive Computing Environment
   - a wonderful programming environment to support the learning of, and use of, Pascal
   - syntax-directed editor knows correct syntax and program structure
   - testing and debugging features
   - built-in help functions
   - fast interpreter (WATCOM Pascal compatible)
   - ready early in 1985
4. Emerald One.....integrated office system
   - for Business Studies, Data Processing, and school administration
   - an integrated package containing:
     - communications tools (electronic mail, telephone message system, interactive "chatting")
     - information handling (relational database structure)
     - decision support (electronic spreadsheet, business graphics)
     - powerful word processing system (Wang-like)
     - time management tools (personal diary, scheduling system)
   - ready for September 1985
5. TURING....U of T programming language that overcomes the limitations of Pascal without sacrificing the Pascal philosophy
   - ready for 1986
6. Logic Gates....logic and circuits
   - the essentials of the modules in Introductory Computer Studies, and C.S. & T.
   - ready for 1986
7. WATCOM Compilers....for BASIC and Pascal
   - useful in all programming courses
   - ready early in 1985
8. ALICE (BASIC)...the ALICE environment (see item #3), but supporting a WATCOM-like BASIC
   - ready late in 1985
9. Database Management System...a sophisticated state- of- the -art relational database
   - available late in 1986
10. DALE....tutorials on how to query and update different types of databases
    - available late in 1986
11. NATAL....an authoring language system that could be used to develop applications in the grade 12 courses, or to demonstrate what non-programmers could use as a software tool
    - available now
    - the CAN authoring system is also being implemented on the ICON (due in 1986)

Additional Computer Studies support software is being developed in conjunction with the Independent Learning Centre. These will be available for field-testing in the fall of 1986, and for widespread use by all Computer Studies teachers in 1987:

i) An inventory control and payroll application package to support the DP curriculum;

ii) A C A D -like tool to support computer technology curriculum;

iii) A computer system simulation in modular fashion to support the teaching of these concepts across the computer studies curriculum;

BEST COPY AVA

iv) Several specialized packages in these areas: sensoring and monitoring simulation, random processes simulation, sorting concepts simulation and an array game to be used in support of the teaching of related computer studies concepts;

v) An assembler language and high level computer architecture emulator for computer studies curriculum.

One of the major concerns addressed by the hardware specifications is software portability. Any software written in the provided languages should be capable of being easily ported to any other O.A.E.M., or be upward compatible with any future models. Thus, the large investment going into software development will be protected. All Ministry-developed software is available to Ontario school boards free of charge: schools have the right to reproduce software and documentation for their own use, as required.

6. CONCLUSION

During the next year school boards will be implementing the new guideline courses, and most will be familiarizing themselves with the ICON microcomputer. New software products relating directly to Ontario's curriculum are being designed to take advantage of the ICON's powerful capabilities, and will be released to the schools as they become available. The year will be one of experimentation, familiarization, and adjustment.

It is an exciting time, and Ontario feels confident in its co-ordinated approach, and confident that its students will be well prepared to meet the changes in the workplace, in technology, and in society.

Kids And Computers - A Summer Program For Elementary Students

Shiela Morrison Cieply
The Board of Education for the City of Etobicoke
1 Civic Centre Court
Etobicoke, Ontario M9C 2B3

Educators do not have to depend on commercially prepared materials to teach computer literacy. Simple and effective lessons can be created by the teacher to meet the needs of individual groups of students. This method is particularly useful to the teacher who posseses minimal skills in the B.A.S.I.C. language, and has limited funds for textbooks and software.

An effective method currently being used to teach skills in keyboarding and programming on the APPLE II and APPLE IIe is the use of LOW-RESOLUTION GRAPHICS. Elementary students respond favorably to keyboarding exercises which result in the reward of a picture on the monitor's screen. After becoming familiar with the keyboard, students move easily into creating their own programs in LOW-RESOLUTION GRAPHICS.

The skills needed to program pictures on the monitor are easily adapted to further computer literacy training for students. For instance, children who see pictorally what a FOR-NEXT LOOP accomplishes can adapt that skill more quickly to a text program than those who have not had this training. In TEXT programming, this concept is sometimes rather abstract for children to grasp. By actually seeing on the screen that a FOR-NEXT LOOP causes a repetition of an action, students can easily transfer this skill to TEXT applications.

Initially, the teacher prepares a set of LOW-RESOLUTION GRAPHICS programs, each of which is built up of simple components. Colors should be grouped in such a way that individual portions of the picture can be deleted for the students to recreate the whole. For example, a seasonal picture such as an Easter egg with a chocolate rabbit, pink eyes, nose and mouth, and green grass can be divided into several parts in order to adapt the program for use with a variety of grade levels. The picture would be saved with only the grass missing for first-grade students. For second- and third-graders, the pink facial features can be left off as well. Fourth-grade students might be expected to type in the instructions for the entire chocolate rabbit, as well as the face and grass. Older students would be knowledgeable enough to type in the whole picture. Thus one lesson is adapted to use with a variety of ability levels.

Once the teacher has prepared the programs and saved the pictures on a cassette or diskette, hard copies of each program should be produced on the printer. One copy of each program list should be made for each computer the teacher has available. The program lists should be mounted on a colorful construction paper or cardboard backing -- cut in a shape which is appropriate for use with the program --- and laminated to protect the lessons so that they might be used from year to year.

When ready for use, each group of students is given one of the program lists. The children in each group are to take turns typing in the instructions in the program -- line by line -- and may RUN the program periodically to see how well they are progressing. This exercise gives them keyboarding practice with a reward, not merely locating keys to match letters or numbers seen on the monitor.

Prior to typing or in the midst of the lesson, the class should discuss what various program lines are instructing the computer to do. New commands can be taught with a ready example within the program to illustrate the command's function. The aforementioned FOR-NEXT LOOP is well illustrated in the process of creating a background color and in various ways as a timing device to allow portions of pictures to come on at varied intervals.

Programs can be set up in such a manner as to allow beginning students to merely type in numbers to create PLOT designs on the screen. This can be accomplished by having the final statements of the program display the command PLOT on the screen and await the INPUT of the numbers needed for each PLOT. A winter design might have snowflakes added by students to a teacher-designed scene. Spring rain can be made to fall on a child in a yellow slicker who is carrying a brightly colored unbrella. PLOTs can become the lights on a Christmas tree or be used to form valentine hearts. There are numerous possibilities , limited only to the teacher's imagination and creativity.

After older students have begun to use teacher - created activities and are familiar with the process of LOW-RESOLUTION GRAPHICS designs programming they will be ready and eager to create their own pictures. Using the students' initials as a beginning project gives them a personal and simple way to design their first picture on the computer. The student's first attempt at designing their initials should be done on a black background with all of the initials done in a single color. This work should first be designed on graph paper, accompanied by a written program which the student has worked out to instruct the computer in the steps needed to produce the initials on the screen. This portion is then typed into the computer and SAVEd on cassette or diskette.

The next step the students must take is to add color changes to the program such that each initial is displayed in a separate color. These additions should be made on the computer after the old program is LOADed into the memory, with the changes SAVEd on the cassette or diskette under the same name used previously.

At this time, the teacher can introduce the FOR - NEXT LOOP and instruct the students in its use for creating a new background color. This addition should be made at the beginning of the program with changes typed in as previously explained and the program should once again be SAVEd under the original name.

A final additional to the program is the use of FOR-NEXT LOOPS to time intervals at which each initial will appear on the screen. This final change should again be typed into the original program and SAVEd a final time. It is a good idea to LOCK the program at this time if you are using a disk drive. The final product is one in which students can take pride and feel a real sense of accomplishment. The entire process takes approximately two weeks when done in one-hour daily sessions with up to thirty-five students and ten computers.

Upon completion of the initial project, students are ready to move into more difficult LOW - RESOLUTION GRAPHICS designs or into developing TEXT program using some of the skills they have learned from this study. In a most enjoyable manner, they have learned a great deal about programming in the B.A.S.I.C. language.

By following these suggested guidelines and utilizing one's own creativity, any teacher can begin to develop lessons to use with their own students. Teachers will find that it is a delightful means of guiding students towards computer literacy after some experimentation on their part.

Identifying Student Characteristics Influencing Success In Introductory Computer Science Courses

Paul W. Oman
University of Odaho
Computer Science Department
Moscow, Idaho 83843

This paper identifies student characteristics influencing success in introductory computer science courses at a small four-year liberal arts college. The primary purpose of this study was to build a simple and practical model to predict success, for purposes of advising students wanting to register into introductory computer courses. Characteristics studied included previous computer experience, number of years since high school, and mathematics and verbal proficiency.

1 Introduction

Current popularity of computers in home and business has led to an increased demand for introductory computer science courses in colleges and universities throughout the United States. In addition to increased instructional loads, this demand places a burden on computer science faculty in terms of advising and placement. The increased volume of students entering computer science coursework has also reduced the amount of time available for individual advising. These trends have created the need for a simple, practical, and reliable model for predicting success in introductory computer courses for the purpose of advising incoming students. Petersen, and Howe (8) reported that students successful in high school mathematics and science would probably be successful in college computer science courses, but no other student characteristics were found to be likely predictors. In a different study, Mazlack (7) concluded that success in college level programming could not be predicted from personal background factors or standardized written tests. Other studies have been designed to identify characteristics of programmers (1,2,9) or to develop instruments for predicting success in computer science courses (3,4,5,6). These instruments typically involve a battery of achievement or aptitude tests which require 30 to 80 minutes to complete and are usually administered to large groups of students. For purposes of advising incoming students in a timely manner, these instruments are not practical in small schools.

What is needed is a quick (less than 15 minutes) predictive model that will identify traits in an individual's background and reliably predict how successful that person will be in an introductory computer science course. This paper investigates to what extent previous computer experience, the number of years since high school, and mathematics and verbal proficiency can be used to develop such a model.

2 Design

2.1 Sample Group

Subjects of this study were students enrolled in introductory computer science courses at Eastern Oregon State College during 1982. The courses were CS 101, "Introduction to Computers," and CS 211, "Introduction to Structured Programming." Both courses have algebra as a prerequisite and are considered to be introductory courses for both freshmen and sophomores. BASIC is the programming language used for the instruction in CS 101, while Pascal is used in CS 211.

In both courses students were evaluated according to their performance on four programming projects, two midterm examinations and a two hour final examination covering social, ethical, theoretical, and historical aspects of computing as well as actual programming concepts. All students attending the classes completed a survey requesting information on previous computer experience and the number of years since graduating from high school.

A total of 60 students were enrolled in the two classes. Selection of only freshman students and elimination of subjects with missing data reduced the number of subjects to 38. The final sample consisted of 20 males and 18 females.

2.2 Independent Variables

Independent variables were limited to information that is readily available to faculty members advising incoming students within a fifteen minute time frame. The following variables were identified as possible contributors to success in an introductory course:

TIMESHARE - the number of time-sharing systems previously used

MICROS - the number of micro-computers previously used

LANGS - the number of programming languages previously used

YEARS - the number of years since high school graduation

MATHSAT - math proficiency as measured by the mathematics Scholastic Aptitude Test (SAT) score

VERBSAT - verbal proficiency as measured by the verbal SAT score

The information on prior computing experience and years since high school was obtained from the survey on previous experience. The SAT scores were obtained through the registrar.

2.3 Dependent Variables

Two dependent variables were investigated as measures of student success in the introductory course:

GRADE - student grade as recorded on an A, B, C, D, F scale and encoded on a numeric scale from four to zero

GRADEPTS - weighted grade computed by multiplying the numeric grade times the credits

3 Results

The degree to which variation in the independent variables relates to variation in the dependent variables was measured by generating Pearson product moment correlation coefficients between all independent and dependent variables. Correlations were also tested for significance from zero. These correlations and associated levels of significance are presented in Table 1.

Table 1

Correlations of Independent Variables
With Dependent Variables
(N = 38)

independent variable	dependent variables	
	GRADE	GRADEPTS
SATMATH	.53 *	.80*
LANGS	.39**	.65*
SATVERB	.55*	.61*
TIMESHARE	.35**	.60*
MICROS	.19	.32**
YEARS	.14	.20

* $p < .01$
**$p < .05$

The independent variable most highly correlated to the dependent variable was MATHSAT followed by LANGS, VERBSAT, and TIMESHARE. These measures held highly significant correlations of .80, .65, .61, and .60 respectively when correlated with grade points. The only other measure correlating significantly with GRADEPTS is MICROS, with a correlation coefficient of .32.

Multiple regression using forward stepwise inclusion was performed with all independent variables, first against the GRADE dependent variable, and then repeated for the other dependent variable. Multiple regression permits the construction of a model using the least squares approximation of the dependent variable that accounts for the largest amount of variance remaining in the model, enters the equation at each step. The best model developed was one using GRADEPTS as the dependent variable. The results of this analysis are summarized in Table 2. The R squared value indicates the proportion of variation within the dependent variable that is explained by the independent variables included in the model at that step.

Table 2

Stepwise Multiple Regression Analysis (N = 38)

WGRADE variable

step	entered	R square
1	SATMATH	.6515*
2	YEARS	.7306
3	SATVERB	.7686
4	TIMESHARE	.8024
5	LANGS	.8216
6	MICROS	.8218

*$p < .01$

The multiple regression model accounts for 65% of grade variation with just the inclusion of MATHSAT (step 1). The subsequent addition of YEARS, SATVERB, and TIMESHARE raises the R squared value to .80 (steps 2, 3, 4). Finally, the addition of the remaining independent variables (LANGS, MICROS) to the model results in a final R squared value of .82 indicating the 82% of grade variation may be explained by this model. An F-test of the model yields a significant F-statistic of 29.484 with 5 and 32 degrees of freedom ($p < .001$).

4 Discussion

The R squared value obtained in this study indicates that much of the variance of performance in introductory computer courses can be accounted for by mathematics proficiency and previous computer-related experience. This model accounts for roughly 82% of grade variability as measured by earned grade points. In contrast, Hostetler's (4) model analyzing cognitive reasoning, personality traits and GPA accounts for only 43% of the variance in final scores. This leads him to conclude that other factors need analysis.

The primary goal of this study was to develop a model useful in predicting success in introductory computer science courses that could be used as an advising tool in a quick and simple manner. The key predictor to success, as determined by this study, is math proficiency. This supports Konvalina, et al., (5) study wherein an aptitude instrument was designed and tested. Their conclusion was that mathematics proficiency is the key to success for computer science students.

The linear relationship between mathematics proficiency and success in an introductory computer course can be used as an advising tool for predicting success. This tool can be improved upon by developing a simple equation yielding a hypothetical grade point derived from the independent variables. By spreading the remaining 18% variation equally across all independent variables the grade point variation is thus assigned percentages of SATMATH - 68%, YEARS - 11%, SATVERB - 7%, TIMESHARE - 6%, LANGS - 5%, and MICROS - 3%. Using these percentages a simple linear model can be formulated that will provide a projected grade to be used while advising incoming students. One such model, applied to two subsequent computer science courses, is correct for 32% of the students, is within one grade either side of the actual grade for 38% of the students, and is off by at least two grade scales for only 30% of the students. Thus, this model can successfully predict a grade for a student, based on information that can be obtained in a 15 minute interview, and be correct or within one grade scale for 70% of the students.

References

1. Alspaugh, C. A. "Identification of Some Components of Computer Programming Aptitude." Journal for Research in Mathematics Education, March 1972, pp. 89-98.

2. Barker, R. J. and Unger, E. A. "A Predictor for Success in an Introductory Programming Class based upon Abstract Reasoning Development ACM SIGCSE Bulletin, February 1983, pp 154-158.

3. Fowler, G. C. and Glorfeld, L. W. "Predicting Aptitude in Introductory Computing: A Classification Model." Association for Educational Data Systems Journal, Winter 1981, pp 96-109.

4. Hostetler, T. R., "Predicting Student Success in an Introductory Programming Course." ACM SIGCSE Bulletin, September 1983, pp 40-49.

5. Konvalina, J., Wileman, S. A., Stephens, L. J. "Math Proficiency: A Key to Success for Computer Science Students." Communication of the ACM, May 1983, pp 377-382.

6. Kurtz, B. L. "Investigating the Relationship Between the Development of Abstract Reasoning and Performance in an Introductory Programming Class." ACM SIGCSE Bulletin, February 1980, pp 110-117.

7. Mazlack, L. J. "Identifying Potential to Acquire Programming Skill." Communications of the ACM, January 1980, pp. 14-17.

8. Petersen, J. M. and Howe, T. G. "Predicting Academic Success in Introduction to Computers." Association for Educational Data Systems Journal, Fall 1979, pp 182-191.

9. Stevens, D. J. "Cognitive Processes and Success of Students in Instructional Computer Courses." Association for Educational Data Systems Journal, Summer 1983, pp 228-233.

Programming Schemes: A Model For Understanding And Teaching Programming, Programming Debugging And Testing

Johnette Hassell, Ph.D.
Tulane University
Department of Computer Science
New Orleans, LA 70118

This session will present a model of how programmers and software engineers learn about and understand programs. Based on empirical results and theoretical concepts, this model, called programming schemes, distinguishes between the abstract concepts of what needs to be done to complete a software task and the concrete representation in code, documentation, or design. The research discussed in this session will: 1) report results on the ability of a concrete implementation to reflect the intended scheme of the author; 2) propose new ways for teaching programming; 3) describe a new classification scheme for errors; and 4) indicate the use of the model in teaching proper documentation.

INTRODUCTION

In teaching students to design and code programs, we usually concentrate on two issues: correctness and form. Students should certainly produce correct programs, however we provide little other than the basic programming constructs for their use in building designs. We expect them to "figure out" how to use these constructs to build up correct algorithms for the assigned problems. Even careful attention to techniques of decomposition does not provide sufficient guidance in techniques for building software.

Typically, as teachers, we do a better job of instructing students in the form code should take. Students learn that variables must have good names, although some research shows that mnenominicy of variable names in unimportant in short programs in the domain of expertise of the programmer (Sheppard 1979). Students also learn the appropriateness of indenting to indicate program structure, upper and lower case letter usage to distinguish reserved words from variables, and comments on the functionality of blocks of code. They often view these standards of form as aberrations of the teacher - much like the English student whose teacher insists on an outline being developed before the theme is written.

An issue little stressed in teaching programming is that all but student programs must be maintained. With two-thirds of the costs of the software industry being devoted to maintenance (Boehm 1976), teachers are remiss in not stressing the importance of creating code and documentation to aid the maintenance programmer who will get the program at a later date - when the original author is unavailable. The situation is not remedied when programmers maintain their own code. A lapse of only a few weeks will render previously ingeniously named variables indecipherable.

An alternate view of the approach to teaching of design and coding lies in consideration of the maintenance progammers. Decisions should made in a manner so that the maintenance programmer can easily learn the intent of the author as well as the functionality of the actual code. This paper presents a model for how maintenance programmers learn about code with which they are unfamiliar. Understanding the model suggests a number of ways to improve the teaching of designing and coding so that it is easier to maintain.

PROGRAMMING SCHEMES

A programming scheme is an abstraction which captures the essential features of a class of solutions to a problem. The abstract concepts are tailored to fit the requirements of a particular situation and the preferences and experience of the programmer. A given concrete instance of a scheme is called a scheme representation. While "file merge" is a scheme which to an expert programmer probably implies the need to acquire and combine two or more ordered files, there can be many different representations for this abstraction. Similarly for other underlying abstractions (schemes), there are many different scheme representations. Thus the familiar conceptual pattern evoked by the term "running count" is a scheme, for which the following are representations:

REPRESENTATION A

```
COUNT <-- 1
while not EndOfInput
  read ARRAY [COUNT]
  COUNT <-- COUNT+1
end while
COUNT <-- COUNT - 1
```

REPRESENTATION B

```
COUNT  <-- 0
while not EndOfInput
  COUNT <-- COUNT+1
  read ARRAY [COUNT]
end while
```

While these examples are simple for concise presentation, schemes and their representations cover a wide range of complexity and difficulty. For further examples of abstract tasks with multiple instantiations see (Hassell and Lind 1983, Nutter & Hassell 1984, and Soloway et al. 1982).

The use of underlying abstractions in understanding has been investigated in other contexts. For example, research on English text comprehension reports that readers use "world" knowledge of the connections between actions and states to detect and correct gaps in stories (Kemper 1982). Reading comprehension is an important aspect of debugging and thus a similar type of knowledge may be used in understanding program texts. The use of abstract conceptualizations of situations has also been identified in other problem solving environments as well (Chase & Simon 1973, Reitman 1976, and Larkin et al. 1980).

The work of Soloway's group at Yale and the author's group at Tulane has demonstrated that programmers similarly use abstract concepts in learning about and understanding a program. See Nutter and Hassell (1984) for a discussion of theoretical issues involving schemes and their representations. In this paper, we wish to discuss how these concepts and findings can be used to teach better programming.

The approach recommended by the author involves planning the program for the maintenance programmer. Thus features that enhance the maintenance programmer's ability to learn about the program quickly and accurately should be the basis for determining quality in programming.

There are characteristics, called distractors, of a scheme representation which inhibit or slow down the learning of the maintenance programmer. These distractors obviously contain poorly named variables, but they also contain other programming characteristics as well. For example, in representation A, the variable COUNT refers to the next available location in the array, rather than the number of items already read. In representation B, COUNT represents the number of items already processed, i.e., the actual count. We say that representation A has a "point ahead" feature since it points to the next available array location. This point ahead feature is common in many algorithms and may be preferred since such algorithms typically have better run time characteristics and better termination criteria.

Another example of a distractor involves loop control. A loop may be constructed so that it depends on a boolean variable such as DONE. The value of DONE is set within the loop body and a maintenance programmer must read those clauses to determine loop termination criteria. Alternately, the loop may state the condition explicitly. In the first case, the use of a boolean and buried termination criteria serve as distractors.

Other categories of distractors have been found and experimental investigations are now going on to determine the degree to which distractors inhibit learning. Additional categories of distractors, examples, and results of the empirical studies will be made available at the conference.

The scheme model provides a new and powerful way to classify errors. A scheme representation's scope is active when its variables are being created or modified and when it is responsible for the program's control flow. The scope is passive when the representation's variables are being referenced but not modified and the representation is not responsible for the program control flow.

In one study on subjects' learning about unfamiliar program code, errors in identification of program bugs could all be classified using the concept of scope (Lind & Hassell 1983). Some subjects correctly found bugs in a deliberately bugged program but continued past the active scope of the representation. In studying familiar code, subjects made significantly fewer errors of this type ($p < ,025$). Other subjects found a partial correction, but failed to follow it through to the end of the active scope of the representation.

Thus, the concept of active and passive scope of a representation suggests powerful ways to improve teaching about original authorship, program maintenance activities, and documentation.

CONCLUSION

The view that programs should be written for maintenance programmers leads to new ways to teaching students to design and code programs. They must learn of characteristics, such as distractors, which will later inhibit maintenance. Additionally characteristics, called triggers may be placed in code to assist in the learning process. The view that a program is made of up scheme representations provides insight about helpful documentation and potential programming and debugging aids. As an extension of the concepts of structured programming, the use of scheme representations to build programs promises improved programs and lesser maintenance costs.

REFERENCES

Boehm, B. W., "Software Engineering", IEEE Transactions on Computers, C-25 (12) pp 1226-41, December, 1976

Chase, W.C., Simon, H., "Perception in Chess", Cognitive Psychology 4, pp 55-81, 1973.

Hassell, J. and Lind, J., "Towards A Plan-Based Strategy for Software Maintenance", Tulane University Technical Report, TUTR 83-104, 1983.

Hassell, J., and Nutter, J.T., "PROGRAMMING SCHEMES: Their Role In Program Understanding and Maintenance", *AEDS Twenty-Second Annual Convention Proceedings*, Washington D.C., 1984.

Kemper, S., "Filling in the Missing Links", *Journal of Verbal Learning and Verbal Behavior*, 21, (1), Feb 1982, pp 99-107.

Larkin, J., McDermott, J., Simon, D., Simon, H., "Expert and Novice Performance in Solving Physics Problems", *Science* 208, pp 140-156, 1980.

Nutter, J. T., "Towards A Theory of Programming Schemes", *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, CO, pp 269-272, 1984.

Nutter, J. T., and Hassell, J., "PROGRAMMING SCHEMES IN PROGRAM UNDERSTANDING AND MAINTENANCE: An Introduction", *AEDS Journal*, to appear.

Reitman, J.S., "Skilled Perception in GO: Deducing Memory Structures from Inter-Response Times", *Cognitive Psychology* 8, 1976, pp 336-356.

Sheppard, S., Curtis, B., Milliman, P. Love, T., "Modern coding Practices and Programmer Performance", *Computer*, 1979, 12 (12), 41-49.

Soloway, E., Ehrlich, K., Bonar, J., Greenspan, J., "What Do Novices Know About Programming", Yale University, Department of Computer Science, Research Report #218, Jan 1982.

An Evaluation Of The Course Content Of A CIS Curriculum

Dr. James Hunter
Western Carolina University
Accounting and Information Systems
Cullowhee, North Carolina 28723

The content of a Computer Information Systems' curriculum has to be evaluated on an on going basis because of the continuous changes that take place in the computer field. Input from CIS employees should be a major factor in curriculum evaluation. This paper presents the results of a questionnaire sent to seventy employees who have hired graduates from the CIS program at Western Carolina University. The questionnaire was designed to give faculty a better insight into what should be taught to CIS majors.

Ever since the Data Processing Management Association Education Foundation presented its model curriculum in 1981, many suggestions have been made as to what should be included in the Computer Information Systems curriculum. Universities, organizations and industries are realizing that the curriculum content should be reviewed on a continuing basis because of the continuous changes in technology. Both hardware and software development are having a tremendous effect on the skills needed to perform successfully in the work place. University professors have long realized that it was difficult to keep abreast of the many changes that have taken place in the industry over the years, and they are changing more rapidly now than ever. In the past schools have been heavily language oriented with emphasis on COBOL. COBOL will probably still be the most predominant language for some time to come but many people indicate that the old traditional COBOL programs are soon going to be a thing of the past and much of the management-decision types of reports are going to be generated through data base query languages.

These new technologies and techniques are having an on-going effect on material taught in CIS programs in order to help keep our department as current as possible, it was felt that we needed input from industry. To satisfy this purpose a questionnaire was developed and sent to the employers of our graduates for the past seven years. The questionnaire was designed for the respondent to rate fourteen areas of knowledge that were felt to be important skills that CIS graduates should possess. An open-ended question was also included where suggestions could be made to include topics that employers felt important and were not included on the questionnaire. From the results of this questionnaire, course content was analyzed to determine if the major emphasis in our courses agreed with what the employers thought was most important.

Companies selected for the survey were those of former students whose addresses were known. The companies represented a good cross section of the job opportunities for CIS graduates. Represented in the group were very large manufacturing companies, computer vendors, university data processing shops, small companies with only two or three people in the data processing department and federal, state and local government agencies. Since very few of our graduates are hired locally there was a wide geographical area representation but the majority of the organizations were located in the Southeast.

The mean score shown for the questions is based on a scale from 1 to 5. Five indicates this skill is essential and 1 means it is not very important. The results of the questionnaire were as follows:

		MEAN SCORE
1.	Have the ability to communicate with technical staff	4.32
2.	Have ability to communicate with non technical people (users, managers, etc.)	4.63
3.	Have ability to design a solution to a fairly complex problem	3.74
4.	Be able to write structured code for a. high level language b. which language ________________	3.74
6.	a. Understand the concepts of how a data base system works	3.79
	b. Have the skills to be able to use a particular data base package	2.47
5.	Understand on-line processing	3.95
7.	Understand internal data representation	3.58
8.	Be able to work with system (canned software packages)	3.05
9.	Have the ability to evaluate software packages in light of special requirements	3.00
10.	Have had experience with micros	3.21
11.	Have had experience working with mainframes	3.63
12.	Be familiar with a particular JCL. Which one __________?	2.53
13.	Have had experience working with others on group projects	3.58
14.	Had on the job training through cooperative education or summer work experience	3.00

Present Curricula:

Before analyzing the questionnaire, our present curricula should be analyzed. Our major's are required to have a general business background which includes 12 courses (36 semester hours) in a business core. The courses include the areas of Accounting, Management, Marketing, Finance, Communications, Economics and one Introduction to Computing course. Besides the business core CIS majors are required to take 7 courses (21) semester hours in the major. Six of these courses are required and one is to be selected from several elective courses. The CIS courses are as follows: (1) Business Computer Programming I-Introduction to programming concepts using a language such as Basic-Fortran, Pascal, (2) Commercial Programming-COBOL, (3) Advanced Programming-Advanced COBOL, (4) Computer Organization and Programming-Internal instruction structure and Organization and architecture of computers including data representation and Assembler language programming, (5) Systems Analysis, (6) Data Base System. Electives: Operations Analysis and Modeling, Operating Systems and Systems Programming, Small Business Computing, Advanced Data Base Design, Software Design, Cooperative Education, Topics in Information Systems.

Our program is very similar to the DPMA model curricula except that DPMA suggest as a required course Structured Systems Analysis and Design and a capstone course for a software project. Faculty responsible for the development of a computer curricula must guard against establishing a set of course requirements then forgetting about how needs change very rapidly. Course content must also be continuously evaluated to make sure it is keeping pace with the material students should know to be able to enter the job market and find and keep a job.

Analysis:

Thirty questionnaires were returned for a response rate of about 40 percent. The results of the questionnaire indicate that communication skills are very important for a person working in the computer field. Even though most business programs require a course in communication, there seems to be a need for CIS majors to have more work in this area. Communication can be incorporated into every CIS course. Students can work in groups in programming courses and the system design courses. Structured walkthroughs should be used in every upper-level programming course. Class presentation should be required in programming courses as well as the systems courses.

Three other areas that ranked high in importance were ability to solve complex problems, ability to write structured COBOL programs and have experience with on-line processing. An Advanced COBOL course could be used to teach these techniques. Complex problems should be assigned in this course with students expected to do most of the work on their own including problems that would require reading and interpreting technical manuals. Problems using on-line processing should be assigned if the software is available.

A data base course should be required but instead of concentrating on using a data base package, it should stress concepts. The theory of how a data base system works is somewhat difficult for students to understand. To help students better understand these concepts, programs could be required that emulate very simple data base concepts. These could involve linked list, inverted files and secondary key processing.

The following question was included in the questionnaire as an open-end question: Please respond to the following comment: Most data processing in the future will be accomplished by packaged programs or through a data base language; therefore, schools should place more emphasis on these than on programming skills.

Some of the responses were as follows:

"Disagree: More future use of package programs in traditional accounting type applications. Expect applications in manufacturing systems and plant automation to continue to be programmed "in-house". Integration of mini-micros in networks with access to, and often controlled by a host mainframe seems to preclude extensive use of package programs. More emphasis should be placed on DB languages, but do not de-emphasize programming skills."

We utilize packages: however, when modifications are necessary you must be able to read and write code!"

"I disagree with the above statement since I believe sound programming skills are and will continue to be a prerequisite to many data processing careers. Also I believe these skills are fundamental to an in-depth understanding of computer technology and its application. I find any argument to the contrary analogous to one's coming to the faulty conclusion that mathematics should not be taught in the public school system, because the calculator or computer (the machine) will do it for you."

"I disagree. While more DP may be done that way, a thorough understanding of analysis, design and programming will remain essential. With a strong background in this, a DP professional can function easily with a canned program and in addition can learn how to modify such a canned solution. All DP shops will continue to need on-site professionals who thoroughly understand programming."

"I agree to a point. The keyword in the above comment is "most". There will continue to be a need for programming skills to edit "packaged programs" or connect them. The computer scientist of the future will need to be a PROBLEM SOLVER in the field of information storage and retrieval. All solutions to problems in this field are not always best solved by using computers as they currently exist. Likewise, packaged programs will not cover every unique problem. Packaged programs and data base languages will not solve industry's problems or satiate the data processing needs. These methods will certainly see increased utilization but only as a set of many tools. These tools when used with discretion and in the proper combination will improve productivity significantly. I see these tools as complements to traditional methodologies including) programming skills and not as total replacements nor as panacea."

Conclusions:

CIS programs are still going to need to emphasize programming skills, at least for the next five to ten years. There are many programs in use today that will need to be maintained in the future. Also most packaged programs need some type of modification which requires a programming knowledge. It will not be sufficient though to just teach programming skills. CIS students will need to have more skills in problem solving, interpersonal relations, and communications. The role of the CIS student will change from strictly programming to that of an "Information Engineer". CIS departments will need to provide coursework and an environment that will enhance technical skills, communication skills, organizational ability and coordination skills.

Using Low-Resolution Graphics To Teach B.A.S.I.C.

Sarah Lapp
Holland Elementary
Madison at Kittle Roads
Holland, Ohio 43528

Etobicoke students don't need distant summer camps to learn more about bits and bytes, witness some 2700 youngsters, all a part of the Summer '84 Kids and Computers program offered by the Etobicoke Board of Education. This will include "how to" information on:

- organization
- curriculum design
- equipment.

In response to the interest of the Community of Etobicoke, the Board of Education for the City of Etobicoke for the past two summers has offered students enrolled in elementary schools the opportunity to participate in a summer computer camp. This paper outlines the computer camp which took place in the summer of 1984.

GOALS AND OBJECTIVES

The program was designed to offer students an introduction to the microcomputer plus a challenging curriculum for the "advanced" student.

Students from Kindergarten to Grade 8 were invited to participate with a quota of approximately twenty-five students from each elementary school. At the final count about 2700 students enrolled in the program.

The camp was offered in three locations - one in the North, one in the Centre and one in the South of the City. These locations permitted students to be in the proximity of their home schools. Anticipating the annual July heat wave, these locations offered air conditioning, minimizing equipment problems. There was no cost to students enrolled in Etobicoke Public Schools.

Three sessions were offered during July and a part of August. Each session lasted two weeks and included four different classes daily. Similar programs were offered on the Apple, Commodore 64 and PET computers. This format was offered at each of the three geographical locations throughout the City.

Students had the option of selecting the session which best suited their needs. The wide variety of choices satisfied most. One program each day was designated as "advanced" while the remaining classes were designed for the novice at the computer.

TEACHING STAFF

Prior to the program's start, teachers were invited to attend an information meeting about the computer camp.

Nearly every teacher who attended, made a commitment to the program. From this group six teachers developed the curriculum for the Apple and Commodore classrooms.

CURRICULUM

Lessons were prepared and distributed to teachers prior to the start of the first session. Each lesson was divided into two sections, the first a lesson plan for the teacher and the second a worksheet for students.

Two teams of three teachers developed the curriculum for both the Apple and Commodore classrooms. Each session included instruction in the operation of equipment, word processing, computer graphics, application software, LOGO and BASIC programming. Lessons were of increasing difficulty and this permitted each student to move through the program at his or her own individual pace.

Software used to introduce the equipment included the Friendly Computer, Apple Introduces Apple, and the PET keyboard. Commands included LOAD and RUN.

Students mastered the keyboard including the alphabetic, numeric and special character keys plus the control keys unique to each machine. Each student mastered some aspect of word processing using The Story Machine, Bank Street Writer, Story Writer or Text Master. Using the printer was introduced at this point.

To achieve facility with the colour and graphics capabilities of the computer, students both wrote programs and used application software. Among those used were the Friendly Computer - The Art Show, Face Maker, Logo, Kidstuff and the Koala Pad. Programming graphics included the principles of VLIN, FLIN, PLOT, cursor controls and the graphics keyboard.

Programming concepts included the immediate mode, mathematical manipulation and the commands NEW, REM, PRINT, GOTO, FOR...NEXT, IF... THEN, INPUT and READ...DATA.

Software from the Board's software library was divided among the three centres and used throughout the program. This included applications in Mathematics, English, Social Studies and the MECC program disks.

Students were not expected to have any keyboarding skills and the basics were presented in order to maximize the benefits of the program. Computer operations and terminology as well as a look inside the computer were all covered.

STUDENT ASSISTANTS

Each teacher was assigned two student assistants. These assistants were computer students from Etobicoke high schools and all demonstrated an ability to work with the elementary school students.

Assistants loaded programs, helped familiarize students with equipment, assisted with application software and with the printing of stories written on the computer by the younger children.

MOTIVATION

Students who attended were motivated by the concept of the program as well as the accessability of equipment. Students were encouraged to write and draw on the computer. Display of their work provided a positive and stimulating visual aid.

EQUIPMENT

Instruction was available on a variety of equipment including the APPLE IIe, Commodore 64 and Commodore PET. APPLE and C-64 classrooms offered individual disk drives while the PET classroom used a network to connect computers to disk drives. Each classroom had eighteen computers and one or two printers. The ratio of student to computer was at least 2:1. Class size varied up to thirty students.

CLASSES

The advanced class was scheduled first each day beginning at 8:30 a.m. Remaining classes were designed for novices at the computer and each session was one and one-quarter hours long. While classes were divided by level of expertise, each classroom included students from all grades in the primary, junior and intermediate levels. The open ended curriculum permitted students of various levels of expertise to move freely through the program at their own pace.

Classes continued throughout the day with a fifteen minute break in the morning and afternoon plus a one-hour lunch break. The day concluded at 3:15 p.m. Classes were not held on Friday afternoons since the caretaking staff operated on a summer schedule at that time.

Some teachers elected to teach two classes per day for a two week session. Others chose to teach all four classes daily for four weeks. Some elected to start the first week in July while others preferred to start after a short vacation. One teacher at each centre was a "Designated Teacher" with responsibility for children who either arrived early or stayed later.

Student assistants continued through all sessions. Because of the overlap of teachers and assistants continuity existed through all sessions of the program.

The program was designed to accommodate 2400 students.however an "overflow" class was created at one centre for additional students who wished to participate in the program.

ADMINISTRATION

The staff of the computer camp offered a variety of backgrounds including teachers of primary. junior. intermediate and secondary school students as well as teachers from the special education program and the administration. The program concept was devised by Gordon McKye, Co-ordinator of Computers in Education, and operated under the auspices of James McMurray, Assistant Superintendent. On a daily basis the three locations were supervised by Summer School Principal, Ken Richards, and a university student.

Technical assistance was provided by another university student and backed up by the Board's Computer Technician. Most equipment problems were repaired with same day service maintaining a full complement of equipment for student use.

In addition to these three locations, an overnight camp was also available to students where one of the programs was, in essence, the curriculum outlined for the daytime camp.

RESULTS

Students participating in the program varied in age from six to fourteen and represented almost every elementary school in the Etobicoke system. During the course of the camp students gained an increased awareness of the computer and how it could assist them in classes such as language arts, social studies, mathematics and art. For many the camp offered an opportunity to meet students from different schools including the secondary schools and to work with different teachers. While some students pursued programming in LOGO or BASIC, others developed an interest in application software such as Oregon Trail. Many became skilled at word processing and graphics on the computer. Each student received a certificate of participation at the end of the two-week session.

CONCLUSIONS

Some students entered the program expecting to be expert programmers at the end of their session. They soon discovered other uses for the computer. Many spent significant amounts of time working with partners to produce imaginative, well-written manuscripts.

Given the large numbers of students enrolled in the program, the impact of the program has been significant. About five students from the summer computer camp can be found in each classroom of our fifty elementary schools. Teachers report an increased computer awareness and classroom curricula have been adapted to meet the new levels of expertise. Parts of the secondary computer program have been mastered by these elementary school students.

Lessons and disks prepared for the computer camp have also had many uses. The lessons including worksheets are available for teachers throughout Etobicoke. Disks used in the program have been duplicated and distributed along with modules of loan computers to forty elementary schools. Disks have been made available to teachers participating in In-Service programs at the Computer Resource Centre.

Based on the concept of student assistants, a co-op credit in computers has been introduced in the secondary schools. Secondary school students can earn a credit while assisting students in a computer environment in the elementary schools.

Academic staff and students benefited from the opportunity of meeting staff and student from different schools, grades and panels offering new insights.

Finally the universality of the computer has increased with both students and teachers finding new and innovative uses for the computer.

Given the success of this summer program, a strong base for computers is now established in Etobicoke elementary schools. It is likely that the computer program will be even more successful in succeeding years.

The ALICE Programming Education System

Brad Templeton and Jim Gardner
Looking Glass Software Limited
3 Menno Street
Waterloo, Ontario N2L 2A4

Abstract

A syntax-directed editor allows programs to be created and edited as logical blocks of code rather than ordinary lines of text. We discuss such editors, and in particular, give a brief overview of the ALICE Pascal system which implements these concepts.

Educators are agreed that students should be taught the techniques of structured programming in any introductory programming course. However, it is often difficult to follow through on this principle in practice. One reason for this is that programs are usually created in a very non-structured way: typed in line by line using a conventional text editor. This leads students to think of programs as a sequence of individual instructions, and they lose the overall picture of a program as a collection of logical blocks.

To correct this problem, a number of groups have developed syntax-directed editors specifically designed to help create programs in a particular programming language. Instead of entering programs in a line by line way, students use syntax-directed editors to construct programs using the logical building blocks of the language. In this way, beginners edit programs as programs, not as arbitrary text material.

As an example of how such editors work, we will talk about the ALICE Pascal system, developed by Looking Glass Software for the ICON computer and other 16-bit machines (ALICE will not run on 8-bit machines because of its memory requirements). ALICE, created for the Ontario Ministry of Education, is one of the first syntax-directed editors to go beyond the prototype stage and enter the full educational software market.

The ALICE system and others like it construct programs using templates, the skeletons of logical blocks of code. These templates contain empty placeholders that the programmer fills in as the program is built. For example, a standard program template in ALICE Pascal looks like this.

```
program name (input,output);
{Comment describing program}
Declarations
begin
Statement
end.
```

The underlined items are placeholders that may be filled in with appropriate program constructs. For example, if a student wishes to begin a program with a Pascal FOR loop (similar to the Fortran DO loop), he or she types "for" in the Statement placeholder. As soon as the editor recognizes the keyword "for", it lays out a template for a standard FOR loop:

```
for Variable := Start to Finish do
begin
    Statement
    end;
```

The programmer then fills in the empty placeholders with appropriate material.

This program construction approach has several advantages. First, the programmer builds the program using the building blocks of the language. The student sees that programs are built with structured blocks, not individual instruction lines. Second, the beginner is not plagued by the countless syntax errors that usually arise from entering a program in the conventional way. Each template laid out by the editor is syntactically correct, so syntax errors like unmatched parentheses, missing "end" statements, etc. simply do not occur. Third, the learner can be led towards good programming habits by building them into the templates of the system. Note, for example, that ALICE Pascal's program template automatically includes a placeholder that should be filled with a comment describing what the program does. Students will automatically absorb many of the basics of good programming style and format.

As a result of all these advantages, students are freed to learn the concepts of programming. Annoying technical details (what bracket goes where) are kept to a minimum.

A Programming Environment:

Once one accepts that syntax-directed editors provide numerous advantages not found with conventional methods of program preparation, the next question is determining the features that should be included in such an editor. A little thought and imagination shows that normal editing is just the tip of the iceberg for such software. Here we have an editor with an intimate knowledge of a particular programming language. Why not add an interpreter for that language? Why not add debugging aids related to the language? Why not add on-line features that provide help on aspects of the language? Why not create an entire programming environment that helps both beginners and professionals create, edit, run, and debug their programs?

In ALICE and some other systems, all facets of program development have been merged into a single complete system. Here are just some of the features included.

(a) Checking for semantic errors, e.g. undeclared variables, operations with mismatched data types, and so on. These are detected as soon as they are typed in, allowing programmers to correct their mistakes immediately.

(b) An UNDO command that allows students to "undo" their most recent editing action(s). If, for example, a beginner accidentally deletes a large amount of source code, issuing the UNDO command will bring the code back again.

(c) The name completion feature allows programmers to type in part of the name for a variable, subprogram, etc., then have ALICE fill in the rest. This means that student programs can use longer, more descriptive names for data objects, without being forced to do a lot of extra typing. Because of the way the name completion feature is implemented, students can also be provided with a list of the names defined in their program (in case they have forgotten some of the names they've defined).

(d) As students are entering a subprogram call, the system prompts for the subprogram's arguments, giving the name and type of each argument. For example, if you are calling the "sin" function, you might see

```
sin( x : real );
```

Again, this means that students will not be tripped up by technical details, and will be free to concentrate on the meaning of their programs.

(e) In order to make programs more readable, blocks of code may be "hidden" so that they do not appear on the terminal screen. Hidden code is indicated with

... {Comment} ...

where the Comment placeholder should be filled with some description of the hidden material.

(f) The system contains an interpreter that will run programs once they have been created. Because of the unique features of the editor, the program is parsed as it is being entered; therefore the ALICE interpreter runs much more quickly than conventional interpreters that work with standard source code text.

(g) Many debugging features are possible. ALICE supports cursor following (having the cursor move through a program as the program is being executed), single-stepping (executing a program one instruction at a time and pausing between instructions so programmers can examine values of variables, test subprograms, etc.), tracebacks (what subprogram called the curren subprogram, what called the caller, etc.), and a number of other standard debugging tools.

Documentation and Help:

All of ALICE is carefully described in a number of documents: a reference manual; an interactive tutorial guide; a complete high school level textbook which teaches programming techniques through ALICE Pascal; and an enormous amount of on-line documentation called HELP files. These files (over 250K of text in total) describe all aspects of ALICE and Pascal, and also cover basic programming concepts. The following kinds of HELP are offered.

(a) Descriptions of all ALICE commands.

(b) Introductory discussions, telling beginners how to use the system.

(c) Lists of Pascal constructs that can be entered in a particular placeholder. Students can select items from such lists and obtain further descriptions. For example, if a student is on a Statement placeholder, this HELP feature would give a list of all keywords and names that could be entered to start a statement. Through a simple process, full descriptions can be obtained for all these Pascal constructs.

(d) Explanations of Pascal features you are using. For example, if you are typing in an array declaration, ALICE can provide you with explanations of arrays, of variable declarations, of declarations in general, or of Pascal programs. The HELP that ALICE offers is directly related to what is being entered.

(e) Expansions of error messages. Normally, ALICE error messages are descriptive, but only one line long. Fuller explanations may be obtained, telling how the error arose and suggesting ways it may be corrected.

Facilities have been provided for teachers to add new HELP information or change existing descriptions, thereby adapting the material to specific class work.

General Operation:

ALICE makes extensive use of menus, thereby reducing the need for beginners to memorize too much when they are first learning about the system. For example, all ALICE commands may be issued from menus. Menu selection is simple: the programmer rolls the ICON trackball to highlight a particular choice, then presses a single key.

To speed the editing process, ALICE commands may be bound to keys or key sequences -- instead of calling up a command menu, you just have to press the right key. ALICE comes with a set of default bindings and the user may define more. Commands may also be typed in directly, without using bindings or menus.

Use in Education:

Syntax-directed editors are in current use in several major universities (e.g. Cornell), where they are commonly used for introductory programming courses (Teitelbaum, 1981). A committee of chairmen of Computer Science departments has suggested that syntax-directed editors are particularly important in meeting the training needs of the coming generation (Yau, 1983).

ALICE is currently being tested with high school students in the Waterloo Region, and will be more widely available by mid-1985. We believe that such a system provides a rich environment for beginning programmers to learn the fundamentals more easily, without being sidetracked by technical details. In the years to come, we expect syntax-directed editors to have significant impact on computer science education, and are pleased that Ontario is in the forefront of these advances.

Bibliography:

Teitelbaum, Tim, "The Cornell Program Synthesizer: a syntax-directed programming environment," Communications of the ACM, Vol.24, #9, pp 563-573.

Yau, Steven, "Meeting the Crisis in Computer Science", Communications of the ACM, Vol.26, #12, pp1046-1049.

SIGPlan Notices, Vol.19, #5, May 1984.

Elementary Computer Literacy: An Inexpensive Strategy

John Good
Central Algoma Board of Education
Box 96, Richards Landing, Ontario P0R 1J0

Central Algoma Board of Education is a small Northern Ontario Board, which like others, was faced with rather severe budget cuts due to the poor performance of the local economy. Both the Board and the teachers recognized the importance of computer technology in the educational system and were able to come up with a very effective strategy to offer computer literacy and computer access to as many students as possible on a limited budget. Projected costs per student contact hour could be under 40 cents in its first year, not including university courses for teachers and interest courses for the local communities!

BOARD CONSIDERATIONS

A computer committee was set up to try to decide both the best microcomputer system and ways and means to implement a computer program for an elementary grade. After a survey of the literacy programs available in other Ontario school systems, we found that their resources were much greater than ours. It was not possible to apply their methods to a system of only seven elementary schools, each a few miles apart. We had to devise a new plan which would suit our own situation.

One idea, which was considered, was to install the lab at a school which is centrally located. Since the Grade 8 students were being bused there for Home Economics and Industrial Arts, the pupils could remain at the school and receive intsruction in computer literacy as well. This however, posed two problems.

1. This would mean the target students would have to be Grade 8.
2. The lab would be idle more than we felt appropriate.

Grade 8 was ruled out as the targeted group, as the Intermediate teachers felt that the curriculum demands would and the time restraints already present in the division would have a negative impact on their program. Since the Junior division allows more flexability in the curriculum, the Grade 6 students were considered to be the ideal target students.

SCHOOL RESOURCES

Each school in the system had at least one PET and one Commodore 64 computer, but the teachers were finding it very difficult to integrate computers into their classrooms with so few machines. If the computer lab was mobile, it would be available to all grades when the computer literacy unit was not being taught. Assuming there would be an adequate selection of quality software, we felt that it would benefit all the students and teachers within the school.

PURCHASES

Since the skills we wished to teach would be dependent upon the system purchased, we found we had to address the hardware problem. The ICON not considered due to its lack of availability

CENTRAL ALGOMA BOARD OF EDUCATION COMPUTER LAB

15	Commodore 64 computers	@ $325.00	$4875.00
15	B&W television sets	@ $84.00	1260.00
1	Commodore 1541 disk drive	@ $325.00	325.00
1	Gemini 10X printer & interface	@ $529.00	529.00
2	Handic Vic Switches	@ $199.99	399.98
6	power bars	@ $19.99	119.94
6	surge protectors	@ $12.95	77.70
300'	4 pair shielded cable & connectors		300.00
	TOTAL --		$7886.62*

Table 1

* plus sales tax

and price. The most reasonably priced computers available to use were Commodore 64's. Unfortunately, buying 15 complete units was too expensive. Table 1 is a list of the equipment we purchased.

THE NETWORK

The Handic Vic Switches are a very inexpensive, but effective method of networking the computers together. Each one will handle up to 8 computers. The host school supplied the 16th computer and the other disk drive. The students were able load and save programs onto only two disks. This not only made the loading of the computers easier, but also collected all student's files onto the two disks. A person using the printer would often 'hang up'the network. The solution to this can be found in an article by Ron Byers, SHARING PERIPHERALS, in the May issue of TPUG magazine. The 'fix' is a simple four pole, three position switch that allows you to switch a computer out of the network to use the printer and then switches it back in. The cost to make the simple device is under $30.

Finding space within a school was and still is a problem. The lab takes up about 20 running meters.

THE PROGRAM

The Grade 6 unit consists of approximately 25 hours of instruction of which about 18 hours is hands on. Since our Board has a rather large percentage of teachers who have received instruction in COMPUTERS IN EDUCATION courses, we have eager and competent instructors.The topics covered in the unit are:

1 - WHAT IS A COMPUTER?
2 - KEYBOARD FAMILIARIZATION
3 - COMPUTERS AND COMPATABILITY
4 - IMMEDIATE MODE PROGRAMMING
5 - COMPUTER TERMINOLOGY
6 - MAKING A COMPUTER PROGRAM
7 - HISTORY OF COMPUTING
8 - CARE AND OPERATION OF DISK DRIVES
9 - WORD PROCESSING
10 - HOW A COMPUTER IS MADE
11 - LOGO
12 - CAPABILITIES OF THE MICRO

By far the bulk of the time is spent on LOGO and word processing. Here, the students use the computer as a tool, which we feel is going to be the most important aspect of computer literacy. Teachers report that these activities are the children's favorite and the quality of their work has improved.

We felt that we should avoid programming in BASIC as much as possible because of the problems the secondary school was facing. They were finding that the students entering the Grade 10 Computer Literacy course had such varying backgrounds in BASIC programming. The teaching of LOGO was seen as good way of challenging the students with a learning type language rather than to teach them the syntax of BASIC.

The previous year we were able to offer a pilot unit with the Grade 6's and therefore felt it necessary to offer these students, who are now in Grade 7, a followup. As a result, they can receive a short unit on LOGO.

TOTAL SCHOOL INVOLVEMENT

Because the lab is in each school for 25 school days, we hoped the other grades in the host school would be encouraged to used it if we supplied a good selection of software. Again, we were faced with a limited budget, so we had to choose the software very carefully. Table 2 lists the software currently available within our system.

Table 2 SOFTWARE PURCHASES BY CENTRAL ALGOMA BOARD OF EDUCATION

Each school has a complete set of:

EDUCATIONAL SOFTWARE CATAOGUE

LONDON BOARD OF ED. programs for PET
(reworked to run on the '64 also)

LONDON BOARD OF ED. programs for '64

A central library has the TVO programs and:

Learning Co.

ADDITION MAGICIAN
BUMBLE GAMES
BUMBLE PLOT
GERTRUDE'S PUZZLES
GERTRUDE'S SECRETS
JUGGLE'S RAINBOW
MAGIC SPELLS
MOPTOWN HOTEL
MOPTOWN PARADE
ROCKY'S BOOTS

Spinnaker

ALPHABET ZOO
KIDS ON KEYS
STORY MACHINE

Commodore

LOGO

Easy Reader

READING COMPREHENSION SKILLS 1,2,3
LEARN ABOUT WORDS 1,2
GRAMMAR
WORLD GEOGRAPHY

Sunburst

CHALLENGE MATH
(THE) FACTORY
SURVIVAL MATH

Timeworks

DUNGEONS OF THE ALGEBRA DRAGONS
SPELLBOUND

Don't Ask

WORD RACE

FUTURE PLANS

The most common complaint about our program is that the computers are not in a school long enough. This has prompted us to look at the purchase of another lab. If this becomes a reality, each school would be able to have at least 16 computers in their school for at least 50 school days.

There will be an expansion of grades reached with computer units next year. This will likey be an extension of LOGO and word processing skills. A Grade 3 LOGO unit, which was piloted this year, will also be available to the students in the Primary division.

Future software purchases will have a major emphasis on courseware. Many of our teachers are asking for programs which will help them teach a series of skills. We also hope to concentrate in other curriculum areas in which we are lacking good software, such as History, Geography, Science and Library skills.

SUMMARY

We are very pleased with both the operation of the lab and the very low cost. It has provided us with the opportunity to have a whole class effectively use computers. This has definately motivated teachers to test the instructional advantages of computers in education.

Acquiring Computer Literacy In Higher Education: A Tennessee Mandate

M. Gene Bailey, Ph.D.
East Tennessee State University
Box 23,830A ETSU
Johnson City, TN 37614

This year, the Tennessee State Board of Regents passed down requirements which would set a mandate for "computer literacy" for all sixteen schools of higher education within its jurisdiction. The purpose of this paper is to present the plan developed and implemented by East Tennessee State University in order to meet the above-stated requirements. This plan is set apart from those proposed by ETSU's sister universities by its interdisciplinary involvement and de-emphasis on programming.

Introduction

The Tennessee State Board of Regents (TSBR) passed down requirements in 1984 which set a mandate for "computer literacy" for all sixteen schools of higher education within its jurisdiction. The requirements apply to all students entering as of the Fall, 1984 term who are seeking either a baccalaureate degree or an associate degree. The TSBR recommended that the term "computer literate" be applied to someone who has the knowledge of and skill to be competent in the following:

1. A communicable knowledge of the history of computers and of the social, ethical, and legal implications and limits of computer use.
2. A working knowledge of computer equipment and technology.
3. The ability to discriminate between problems that can and those that cannot be appropriately solved on a computer.
4. A modest capability in a programming language.
5. The ability to use the computer effectively in educational and career tasks.

East Tennessee State University's Policy on Computer Literacy

Effective with the academic year 1984-85, all entering students -- those studying for a baccalaureate degree and those studying for an associate degree designed for transfer purposes -- must demonstrate a level of computer literacy equivalent to a knowledge level comparable to that required in general education courses. The adopted definition of computer literacy was that proposed by the TSBR and given above. The policy will apply to students as follows:

1. Students who are seeking an associate degree in curricula designed for transfer must meet the computer literacy requirement prior to their graduation. Baccalaureate students must satisfy the requirement within the first sixty semester credit hours of study.
2. Students transferring into academic study at ETSU with sixty or more hours of credit must meet the computer literacy requirements within

their first thirty hours of study at the university. Students transferring in with less than sixty hours of credit must meet the requirements by the time they have earned sixty hours or their first thirty hours at ETSU, whichever provides a longer period.

Proficiency in the required competencies may be achieved by either

1. An approved course or portion of a course, or transfer of a course equivalent to an approved course.
2. Validation of proficiency through successful performance at a level of 70% or higher on a challenge exam, designed by the Department of Computer and Information Sciences and approved by the Computer Literacy Review Panel, to measure the five competencies identified above.

A Computer Literacy Review Panel was established and its members named by the Vice President of Academic Affairs. The panel includes one faculty representative from each school /college, one academic dean, one representative from the Office of Admissions and Records, and one undergraduate student representative. This panel is charged with the responsibiliy of:

1. Reviewing all proposals for existing courses which schools/colleges wish to have approved for meeting the computer literacy requirement as described and making recommendations to the Academic Council for final approval.
2. Approving petitions from students who wish previous experience or non-credit study to qualify as meeting the computer literacy requirement.

Finally, all students must submit documentation to the Office of Admissions and Records prior to completion of thirty semester hours of study, indicating either completion of an approved course, successful performance on the challenge examination, appropriate transfer credit, or plans for attainment of the above under the time restraints given.

ETSU's Undergraduate Course in Computer Literacy

The material for the course is divided into two categories; theory and "hands-on". Students spend half the semester studying topics related to competencies 1), 3) and 5). A major emphasis is placed on terminology. This part of the course is taught by a member of the Department of Computer and Information Sciences in a lecture environment accomodating approximately one hundred students. The topics include history, terminology, storage devices and file processing, input/output devices, data communications, security and privacy, and computers in society.

The second half of the course is designed to give students "hands-on" experience. Twelve IBM Personal Computers were purchased specifically for this course. Students work in pairs, allowing twenty-four to enroll for a lab. As enrollment increases, the labs will extend to include the three other microcomputer laboratories on campus (one other IBM and two other APPLE labs).

The laboratory portion of the course is designed as an interdisciplinary course. Three primary software applications are studied; wordprocessing, databases (filing systems), spreadsheets. A minimal amount of time is spent examining BASIC. The lab sections are taught by faculty from other departments. The different topic areas are covered but with specific applications in the interest area of the instructor. Students will eventually have an opportunity to enroll in a lab section that will discuss applications that will best suit his/her individual needs.

Advancements in computer technology have permitted the development of software that is very easy to use. As a result, it is no longer necessary for the ordinary user to be proficient in, or even knowledgeable about, programming. In a paper presented at the ACM National Computer Conference, February 1984, Dan McCracken[1] cautioned against teaching programming to individuals who were not planning on a career in computer science. Dr. Louis Robinson,[2] Director of University Relations for IBM Corporation, defines computer literacy as the ability to use computer technology in a discipline. He feels that it does not imply an accomplished capacity to program. These are only two of a growing number of experts who are de-emphasizing programming for the non-computer-science major.

The emphasis, then, of the Computer Literacy course at ETSU is placed on the use of generally applicable software packages. The desired effect is that students will learn to use the computer as a tool throughout their education and career. This exposure should give them a basis for learning to operate other packages easily and usefully.

Expected enrollment for this course is approximately two hundred per semester. The theory and lab portions will run concurrently with half of the students attending lecture and half attending lab. Midway through the semester, the two groups will exchange, thus allowing both groups equal exposure to theory and applications.

Initial Problems - Future Research and Development

The first course in Computer Literacy was announced just two weeks prior to commencement of the fall semester. As a result, only forty students enrolled for the course. A decision was made to teach both theory and applications for the entire semester, instead of concentrating on each of the areas for half of the semester. This decision was made because of the lack of time necessary to develop the laboratory manuals and assignments. Students spent 1 1/2 hours in lecture and 1 1/2 hours in lab each week.

The results of such a split were very negative. The students indicated almost unanimously that the short time in lab each week created difficulties in learning the software packages. All felt that the applications packages would best be learned in a "dedicated" environment - where students would concentrate only on the "hands-on" portion of the course each week, but for only half the semester. As mentioned earlier, the original plan for the course intended such a division. This fact, encouraged by the students' comments, resulted in scheduling the course as originally planned for the second semester.

The implementation of ETSU's Computer Literacy requirement has exposed other unforseen problems which will require additional study before they can be resolved. The problems seem to be primarily in two areas:

1. The logistics of lab scheduling and of assigning faculty to these labs.

2. Establishing a policy of awarding credit to faculty members of other departments for instructing the various lab sessions for the computer science department.

In addition to the problems listed above, others have surfaced in attempts to implement the alternate means for meeting the requirement. In particular, no guidelines exist for evaluating petitions for credit based on experience and/or other courses of study.

The Computer Literacy Review Panel and the Office of the Vice President of Academic Affairs are working on possible solutions to the previously stated problems. The recommendations of the panel for solving these problems and any new ones which may arise during the first year of the computer literacy requirement will be presented in a later paper.

Conclusion

The Computer Literacy Requirement established by the Tennessee State Board of Regents was mandated with only the best of intentions. However, their stated definition of computer literacy is quite vague. In particular, the "modest programming proficiency" requirement has been interpreted in eighteen different ways - one for each of the State Universities and Colleges in Tennessee.

At the Annual Southeast Regional Meeting of ACM in Gatlinburg, two of the sessions were devoted to computer literacy. Many institutions (from both within and without Tennessee) presented their plans for instituting the requirement and a description of the courses involved. There was general agreement in the three topical areas of wordprocessing, databases, and spreadsheets. However, very few agreed in the quality and quantity of programming that should be taught. The courses offered varied from one-hour crash courses in terminology to three/four-hour intensive programming courses.

At issue, also, was the TSBR's policy that the course be a requirement rather than an elective. The addition of a required course has played havoc with the general education requirements of every school in Tennessee. If the number of hours that students take for graduation is not to increase, what subject should be replaced by computer literacy?

A general feeling of panic has arisen due to the rapid growth of microcomputers in the market, and perhaps, the brakes should be applied to give institutions (administration, staff, and faculty) an opportunity to review very carefully the entire issue of computer literacy.

ETSU is making every effort possible to meet the growing and changing needs in the field of computer literacy. It has been determined that through the cooperation of all departments within the university, a sound policy of computer literacy can be developed and implemented to meet the requirements of students enrolled in each discipline. Since reluctance exists to include computer literacy in the general education requirements, departments are being encouraged to include computer applicatons in existing courses. This would allow students an opportunity to use

113

the "computer as a tool" in their real areas of interest. If such a plan were successful, there would be no need for special courses designed to make students "computer literate."

References

1. Proceedings of the ACM Computer Science Conference, Feb. 1984.
2. "Computers in Education: Implications for Schools and Colleges," Regional Spotlight, Jan. 1984, p.3.

EDUCATIONAL TOOLS

Authoring Language/Systems In K-12 Schools: What's Happening

Michael L. Burger
Texas A&M University
620 Harrington
College Station, Texas 77843

In 1984, attempts were made to identify K-12 educators across the United States who were using authoring languages/systems (AL/S) to determine how many teachers are using AL/S, how AL/S are being used by these teachers, and to assess how these teachers rate the importance, extent of use, ease of use, and enjoyablility of each of fourteen features common to many AL/S. Only twenty teachers were identified by procedures used in the study. Lack of incentives, the length of time required to develop CAI using AL/S, and AL/S which have features missing or difficult to use seem to be deterring widespread usage of AL/S by K-12 educators.

Authoring languages and systems (AL/S) are being marketed to K-12 educators with hopes that they will help teachers write their own CAI. The literature has not addressed questions of whether or not teachers are finding AL/S useful and are using them for software development. AL/S will be used in K-12 settings if they can easily be used to create quality CAI and if sufficient incentives exist which encourage the use of AL/S.

In 1984, attempts were made to identify teachers in K-12 settings across the United States who were using AL/S to find out how many teachers were using AL/S, how AL/S were being used, and what features were considered to be important by teachers who use AL/S. One hundred twenty-nine letters were sent in February, to presidents of state educational computing organizations, consultants with state and county offices of education, and directors of intermediate educational associations. Sixteen responses were receivedfrom this proces, three of which consisted of form letters providing organizational and membership information. One respondent indicated that she knew of no one in her state who was using AL/S, and noted that she had forwarded the letter to a colleague (who did not respond).

These people were asked to help identify schools and/or teachers which were using AL/S.

Using this process, sixty-seven educators in fifteen states were identified who were either using AL/S or knew of someone who was. Early in April, a letter and questionnaire was sent to each of the teachers who were identified and twenty-two responses were received, two of which were not usable. A follow up letter was sent the first week in May, 1984 to the individuals who had not responded to the survey item, but no additional surveys were returned. Results were compiled from twenty responses from K-12 teachers in four states (California, Kansas, Minnesota, and Montana) who were using AL/S at the time the study was conducted.

In the Fall of 1984, in an attempt to raise the number of participants in the study, another attempt was made through the Texas Computer Education Association (TCEA) to identify users of AL/S. No additional teachers had been identified as a result of this last effort at the time this paper was written.

Eight different AL/S were represented in the twenty responses received. Four of the respondents indicated that they received incentives for developing their own CAI. Two received extra pay, one (a resource teacher) said it was "fun", the other did not indicate what type of incentives were received. AL/S were most heavily used in the language arts area, but were also used (in order of frequency) for math, computer literacy, science, social studies, vocational education, reading, and special education classes. There was a balanced use of AL/S from the fourth grade through the twelfth grade, but only two respondents indicated use in kindergarten classes.

Cost and features were cited as the most important reasons for selecting the authoring languages. Of the fourteen features listed on the questionnaire (see Table I), however, five (graphics, animation, branching, sound, and interactive video) were identified by at least one half of the respondents as not available on the AL/S they were using. When asked what features should be added to their AL/S, student records and graphics were mentioned the most, branching and animation were the next most frequently mentioned features.

Respondents reported spending an average of seven hours and eighteen minutes developing lessons, but the median development time was two hours. The average and median length of time students spent working the lessons created was twenty minutes. This compares with data in the literature which suggests that from twenty-five to four hundred hours are spent developing one hour lessons (Pogue, 1980; Osin, 1976). Some authors spent as little as thirty minutes developing short (five to ten minute) lessons, others spent over forty hours developing twenty minute lessons.

A comment received from six of the twenty respondents, was most strongly stated by one teacher who did not complete the survey. He wrote "Creating lessons takes too much time and effort for the educational gain. I could not think of, or send you the name of, one teacher who is still using an authoring system. They have all given up after a few years." That the procedures used in the survey turned up so few respondents cannot be totally explained by lack of appropriate research design. The appearance is that very few K-12 educators were using AL/S in 1984.

None of the respondents had worked with an AL/S more than three years. Fifty percent had used the AL/S one year or less. Thirteen of the twenty respondents trained themselves how to use the AL/S, two with the additional aid of a workshop.

Respondents were asked to rate the importance, extent used, enjoyability, and ease of use for each of fourteen features on a five point scale (one being low, and five being high). Each feature's rank was determined by calculating its average rating among the respondents. The features were placed in order according to their average scores. Table I shows the rank order of each of the features listed on the questionnaire. Features listed at the top of the table received the highest ratings.

Because of the limited sample size, no detailed analysis of the relationships of the rankings of the features between columns was done. But, examining the relative positions of a given feature across the different columns yields interesting information. For example, several features were ranked considerably lower in the

TABLE I

Rank	Importance	Extent Used	Enjoyed Using	Ease of Use
1	Editing	Lesson Editing	Student Records	Class Roster
2	Student Records	Security	Security	Security
3	Random Questions	Student Records	Interactive Video	Sound
4	Security	Text Display	Response Timing	Student Records
5	Text Display	Random Questions	Text Display	Text Display
6	Branching	Branching	Class Roster	Editing
7	Item Analysis	Documentation	Editing	Random Questions
8	Class Roster	Graphics	Random Questions	Response Timing
9	Documentation	Response Timing	Item Analysis	Item Analysis
10	Graphics	Interactive Video	Sound	Branching
11	Sound	Sound	Documentation	Interactive video
12	Response Timing	Class Roster	Graphics	Documentation
13	Animation	Item Analysis	Animation	Graphics
14	Interactive Video	Animation	Branching	Animation

"enjoyability" and "ease of use" columns, than they were in the "importance" column. Editing, random questions, and branching, all rated as important features, were not rated as being as easy to use or as enjoyable to use, as were other features which were considered to be less important (e.g., response timing, and class roster). If the more important features could be made as easy and enjoyable to use as are some of the features which are not as important, perceptions of the usefulness of AL/S in K-12 settings might change.

Interactive video was rated the least important of all of the features, but very enjoyable to use. This could be explained by research which suggests that often, new technologies are used because they are novel, and not because they have been found to be important elements in instruction. Another explanation is that teacher-authors, generally, have not used interactive video techniques and don´t know how important it can be.

The benefits received by teachers from developing their own CAI, at the time of this study, did not appear to be significant enough to encourage widespread use of AL/S by K-12 educators. With the limitations of present AL/S systems, and the lack of external rewards (e.g., released time or monetary compensation), educators who are likely to use AL/S, will use them to develop CAI that is simple to create and which is limited to the use of simple instructional strategies. Results of this study, as well as conclusions found in the literature, suggest that AL/S which offer a great number of features and allow the creation of complex lessons involving complex instructional strategies do not allow easy development of complex CAI. Apparently, the K-12 educators who use AL/S will create drill and practice type CAI and will base their CAI strategies on the "instructional strategies" that are built into the AL/S as defaults.

Conclusions

Given the limitations on time and other resources which typically restrict the activities of K-12 educators, widespread authoring of courseware in these schools is not likely to occur using present AL/S. None of the AL/S examined in this study, could be easily used by these educators to create CAI that is more sophisticated than drill and practice. The development of lessons which contained more complex instructional strategies is clearly impossible with some of the AL/S, and extremely time consuming and difficult with most of the others. To the extent that some teachers (using present AL/S) will be able to effectively develop drill and practice (and to some limited extent, tutorial) CAI, and to the degree that those individuals find that process rewarding, a limited market for AL/S exists in K-12 settings. AL/S will seemingly be more extensively used in software development operations and in CAI development team efforts in other settings.

Therefore, AL/S which are designed to be used in K-12 settings should:

1) be based on the features which the teacher/author is most likely to use, and should provide information regarding the limitations and appropriate applications for CAI developed with the authoring software. Instead, unreasonable claims suggesting that the AL/S can be used to teach anything to anyone are often found in the advertisements and manuals.

2) provide authors with fast and easy ways to enter, test, and revise all or selected parts of their lessons. The time involved just switching between developing and testing (if too long) can serve as a deterrent to the creation of good CAI and to the use of the AL/S at all. For example, a few AL/S allow authors to move from editing to testing in less than ten seconds, while others require more than two minutes.

3) include features that allow authors to obtain feedback on the performance of their students. AL/S which assist authors to evaluate their CAI lessons and their students' performance can serve as an added reason to choose to work with the authoring package. Simply having such capabilities, however, is not enough. AL/S need to provide the information most likely to assist the author/teacher in judgments he or she wants to make about the students and the lesson. Additionally, the gathering and reporting of that information needs to be easily accomplished.

4) provide inexpensive ways for several teachers in each school district to obtain copies of the AL/S (lesson creation and lesson presentation modes) and encourage exchange of authored software between teachers in different school systems.

References:

Chambers, Jack. & Sprecher, Jerry W. (1980). Computer assisted instruction: current trends and critical issues. Communications of the ACM. 23(6), 332-342.

Kearsley, Greg. (1982). Authoring systems in computer based education. Communications of the ACM. 25(7), 429-437.

Osin, Luis. (1976). SMITH: How to produce CAI courses without programming. International Journal of Man-Machine Studies. 8, 207-241.

Pogue, Richard E. (1980). The authoring system: interface between author and computer. Journal of Research and Development in Education. 14(1), 57-68.

Roblyer, M. D. (1983). Toward more effective microcomputer courseware through application of systematic instructional design. AEDS Journal. 16(1), 23-32.

Zinn K. L. (1974). Requirements for effective authoring systems and assistance. International Journal of Man-Machine Studies. 6, 381-400.

Second Generation Electronic Spreadsheets And The Teacher's Gradebook

Samuel W. Spero
Cuyahoga Community College
2900 Community College Ave.
Cleveland, Ohio 44115

An electronic spreadsheet program such as Supercalc-3 is introduced to be used by teachers in the preparation of their grade books. The technology to run these programs is relatively inexpensive and presently available in many schools, colleges and universities. These second generation spreadsheet programs can be used to sort information in the gradebook, to prepare report cards, to prepare reports from the data within the gradebook, and to graph the information within the gradebook.

Introduction

The electronic spreadsheet is a computer program that permits you to use your computer screen as a large sheet of paper and the keyboard as a pencil and calculator.

The electronic spreadsheet is ideal for use as a teacher's gradebook. Information can be conveniently added or deleted. Statistical information by student and by test or quiz can be easily calculated, and different grading schemes can be tried with virtually no work. Furthermore a printout of this information can be had by typing in a simple command.

These applications have been described elsewhere, e.g. VISICALC HOME AND OFFICE COMPANION by David Castlewitz and Lawrence J. Chisausky with Patricia Kronberg, page 126 .

Second generation spreadsheet programs include additional capabilities. These programs can sort the information within the spreadsheet in many different ways. They can manipulate the information within the spreadsheet so that subsets of students or tests can be selected out for special reports. They include the ability to prepare report cards for each individual student. Different graphs of the information within the spreadsheet can also be provided. For example, histograms, pie charts or x-y plots can be drawn very easily by these programs so that the teacher and their students can view the information within the gradebook visually as well as in tabular form.

The Electronic Spreadsheet

(In this paper we will be using the electronic spreadsheet program called SUPERCALC-3. The concepts presented do apply to virtually all of the major second generation electronic spreadsheet programs with only minor modifications.)

The spreadsheet may be thought of as a grid. Columns in the spreadsheet are identified by letters of the alphabet and the rows are identified by numbers. The earliest spreadsheet programs like VISICALC included 63 columns and 254 rows. The newer spreadsheet programs like SUPERCALC-3 and LOTUS 1-2-3 include even more rows and columns. The intersection of a row and column, designated by the column letter and the row number, e.g. B24 is the intersection of column B and row 24, is called a cell. All activities on the spreadsheet take place within a cell.

On the spreadsheet there is one cell which is highlighted. This is called the active cell. Whatever is typed on the keyboard is entered into the active cell. The active cell can be moved from place to place using the arrow keys which permit movement up, down, to the right and to the left. The position of the active cell is identified by a cursor. When the cursor encounters a screen boundary the spreadsheet will "scroll."

The Electronic Gradebook

To enter alphabetic information the active cell is moved to where the information is to be entered. A quote followed by whatever is to be entered onto the screen is then typed. The information is entered into the active cell when the ENTER key is pressed. For example, the title of the gradebook for a Mathematics course I have been teaching is:

MATHEMATICS 100 - SUMMER, 1984

Starting with a quote, this phrase is typed onto the keyboard. Pressing the ENTER key moves the title into the active cell.

Other information about the course as well as the names of the students and the labels above each column into which data will be entered can be entered in the same way.

Numbers can be entered directly without quotes. The students grades over the course of the quarter are entered in this way.

In addition to text and numbers, formulas can also be entered. The variables in the formula are the cell coordinates of the values that would be substituted into the formula.

For example, the formula for calculating the average of all of the grades might be calculated as follows:

```
.8*AVERAGE(B10:G10)+ .2*H10
```

where the * indicates multiplication and the + indicates addition. The grade is calculated on the basis of 80% of the grade being contributed by the tests which are contained in the cells B10 through G10 and 20% of the final grade being contributed by the final examination which is in cell H10. The AVERAGE function calculates the average of the numbers in the row of cells from B10 to G10 (the : indicates a range, i.e. from B10 to E10). In addition to the basic operations of addition, subtraction, multiplication, and division SUPERCALC-3 has available an elaborate set of mathematical functions available to the user. AVERAGE is just one of these.

To facilitate the typing of information which repeats information already typed, in this case the formula for the determination of the final average in the course, the REPLICATE command is used.

The REPLICATE command asks you to first identify the cell in which the formula you want to replicate resides. It then asks you to identify the range of cells into which you want to replicate the formula. Since the formula is stated relative to the row or column in which it is typed, the computer will automatically adjust the formula to take this into account. However, there is an option that you can select which will permit you to keep any or all of the variables constant.

Applying the REPLICATE command to the above grade sheet the final averages are computed automatically. The command in SUPERCALC 3 appears as follows:

```
/R(eplicate),I10,I11:I30
```

where the final grade for the first student is in I10, the grades for the other students are in I11 through I30. The command says replicate, relative to the row in which it appears, the formula in row 10 into rows 11 through 30.

Since it requires so little extra work averages for each of the tests can also be very conveniently calculated.

What is important to appreciate is how little background is required to prepare the above.When this is compared to the knowledge required to program the gradebook in BASIC or PASCAL, the differences are considerable. Since the computer is meant to be a tool for the teacher and not an end in itself programming should not be a necessary skill for the classroom teacher. Yet the classroom teacher should not be restricted to some "canned" gradebook program which restricts the flexibility inherent in the electronic spreadsheet.

Second Generation Features

The first of the additional features is the ability to sort. This is particularly valuable to the classroom teacher who might want to rearrange his or her gradebook in some other order. For example, if the teacher wants to post grades in the order of achievement the teacher would type in the ARRANGE command as follows:

/A(rrange),c(olumn),F,10:40, A

where the teacher only has to type /A (the computer fills in the rest of the word).The teacher is then asked for R(ow) or C(olumn). The particular row or column is identified (in this case the column with the final grades) and then the teacher specifies the rows which are to be sorted (rows 10 through 40) and their order (A(scending) or D(escending)). Pressing the ENTER key then immediately performs the sort. This new spreadsheet, or whatever part of it is desired, can then be printed out and posted. If the teacher uses identification numbers, the column with the identification numbers and the final grade column can be moved to another spot on the spreadsheet and printed out.

A second feature is the preparation of "report cards" for each individual student using the /X(excute) command. We prepare a spreadsheet which contains the commands we want to be executed to prepare a report card for each student, and we save this spreadsheet as a PRINT file. When we execute the /X command it asks for this file and then immediately begins to prepare each report card. Once this command file has been prepared it could be used by the teacher to prepare a weekly report for the students. The same Execute File could be even be used for different grading periods. Once the effort has been expended in the preparation of the Execute File it can be used repeatedly with little effort on the part of the teacher.

Besides report cards the classroom teacher is also often asked to prepare other types of summary reports for the school administration. The data management capabilities of the second generation programs offer significant assistance to the classroom teacher in this area as well. For example, the school administration might ask the teacher to prepare a list of all of the students performing below or above a certain level (the honors list or the deficiencies list). To prepare this list the teacher could use the //D(ata) command.

The last of the features we will discuss is that of the graphing capability of the second generation spreadsheet programs. For example a histogram (or bar graph) of each of the tests and of the final tests and the final grade is a desirable report for the teacher's files. These can be generated very quickly either on the screen or on the attached graphics printer complete with headings. The same type of graph can be prepared for each student with reference to their performance over the course of the quarter or semester so that patterns of achievement can be easily spotted. A pie-graph would illustrate to the student (and their parents) how the final grade is calculated and what portion of the student's achievement contributed to the student's final grade. An X-Y graph would illustrate trends in achievement which could be very important in future planning. The commands that would be used to prepare the graphs are subsumed under the /V(iew) command.

Discussion

The gradebook is the product of one of the important aspects of the role of the teacher - that of administrator and evaluator. In this role the teacher must perform an important but nonetheless tedious amount of data processing. To the extent that this is only one of the functions of the teacher, the busy work is relegated to a tertiary position and is very often not done or if done it is done in a minimal fashion with no real attempt made at evaluating the information being presented. Enter the computer. The data processing is what the computer was designed to do and as we have shown it does it well and in a fashion to provide maximal help to the teacher with the minimum investment of time. Using the computer teachers can at last spend their time evaluating and not just data processing.

A final point which should be mentioned. To use this gradebook program it is not practical for the teacher to rely on a computer available only in the school. We are looking toward a near future when each teacher has their own computer at home so that when they finish grading their students' work they can immediately enter the grades without first writing them down on paper and then transferring them to the computer the next day or at some later time. That is extra work and will militate against the computer ever being used to help the teacher. Only when the computer is conveniently accessible will significant changes take place in teaching practice.

Many teachers are buying personal computers for their homes. Apropos to this paper, the second generation spreadsheet capabilities are readily available on the IBM PC Junior, an excellent and relatively inexpensive personal computer that offers many advantages for the classroom teacher.

Computers and Composition In the Music Education Classroom

Byron Hermann
The Ontario Institute for Studies in Education
Toronto, Ontario

This paper reports the findings from a study investigating the cognitive requirements of music composition and the use of the computer as a tool for teaching composition in the classroom. Also under investigation was a teacher intervention strategy known as *procedural facilitation* as a means of facilitating these identified cognitive processes. Results of the study show that both the use of the computer music system and the use of procedural facilitation offered assistance to the subjects involved in the revision process.

INTRODUCTION

As new innovations resulting from technological advances are considered for educational purposes, creative educators attempt to adapt the new tool to meet specific needs or to fill gaps that have been perceived in available resources. One such gap can be found in the music education curriculum, more specifically in music composition related activities. This paper will look at how microcomputer-based computer-music systems can help fill this gap by providing a valuable tool for use by students for this creative activity and by music educators in teaching the concepts of music composition.

Very few attempts have been made to develop computer-music systems for instruction in music composition. There are many music editors, such as the ALF Music System and alphaSyntauri's *Simply Music* to mention just a few, which could be adapted as a tool for composition instruction. Two music systems in particular, the Logo Music system and Musicland Games, are designed to facilitate the development of compositional skills. Upitis (1982) has investigated the potential of using the ALF system for the teaching of composition and reports favourable results. However, no studies could be found which discuss the use of computer-music systems for teaching the *processes* of music composition. This was one of the reasons for undertaking the study reported in this paper.

DESCRIPTION OF STUDY

Two experimental studies were carried out involving subjects from one grade 7 and two grade 8 classrooms using the alphaSyntauri *Simply Music* computer music system as a tool. Experiment 1 investigated the effect of teaching concepts related to timbre with full or partial use of the computer tool compared to a control group. Full use of the computer tool included demonstrations by the instructor of concepts such as sound wave construction and music envelopes with the assistance of the sound editing capabilities of the alphaSyntauri software as well as allowing students to experiment with the actual creation of various timbres by manipulating the sound wave and envelope parameters. Experiment 2 investigated the revision process in music composition. Selected students completed a music composition task either with or without the use of the computer music tool and either with or without the use of a particular intervention known as procedural facilitation. During the course of a private interview, students were asked to "think aloud" while composing and the verbal protocols were transcribed and analyzed.

The technique of procedural facilitation is derived from work in English composition done by Scardamalia and Bereiter (1983). This technique offers some facilitation of one or more of the executive routines required in the composition task. In this way the student can direct more attentional resources towards the production of the material. Procedural facilitation provides a way for the underlying compentencies of the student to appear as the resource demands of the executive system are reduced. In this experiment, evaluative and directive phrases are presented to the student by the computer for consideration during the composition revision process as a method of facilitating the comparison of the intended sound representations and the actual sound representations.

COGNITIVE PROCESSES IN COMPOSITION

Cognitive processes in composition can be described in terms of planning, problem-solving, and decision-making activities. The music composition process is a process of moving from an initial musical idea to a completed unfolding of that germinal thought. The source of such initial ideas is often a matter of creative inspiration. however, such musical ideas can often be pulled out of a child's creative mind with the assistance of a descriptive, programmatic atmosphere, be it a mood, a thought, a scene or a dramatic situation.

Once the creative process has been set in motion, the composer is faced with a series of decisions based on a large amount of information. These decisions involve the organization of sound stimulus data into meaningful musical units. Such decisions are made based on two main sources: the composer's accumulated knowledge of the language used to express musical ideas, and the composer's sensitivity of musical expression.

As the process continues, the specificity of the decisions keeps increasing, right down to the smallest detail. Upon completion of initial musical units or initial drafts of entire compositions the decision-making procedure continues as a revision, or series of revisions may be required. If at some point the composer is satisfied that the finished product adequately reflects his initial intentions, or that he cannot improve on the product, however dissatisfied he may be with it, then a final decision is made to end the process.

Three particular cognitive requirements can be identified from this description of the composition process:

1. Accumulated knowledge structures or compositional building-blocks

2. Planning

3. Revision

COMPOSITIONAL BUILDING-BLOCKS

One of the goals of music education is to provide the students with an understanding of the basic elements of the musical language, the syntax of that language, and the application of that language for self-expression. Children involved in composition create music out of the elements which they possess, reflecting whatever stage of musical knowledge and development they have reached. These knowledge structures, which provide the raw material for the composition process, have been termed compositional building-blocks. The results from experiment 1 show that students who had full use of the computer music tool for the instruction of concepts related to timbre, a compositional building-block, demonstrated a better understanding of those concepts than students who had received only a "traditional" instruction in the subject matter. Experiment 2 investigated various aspects of melody compositional building-blocks as displayed through the compositions of the subjects. Results show that the overwhelming occupation of student mental activity was in relation to contour, followed by concerns with goal-of-motion or the tonic pitch. Little consideration was given to structural concerns. This result can be explained in terms of the level of planning which was involved as well as the type of mental processing required. These low level concerns seem to be closely linked with the overall musical sensitivity of the student and seem to require little mental attention.

PLANNING

The cognitive requirements of planning can be categorized in terms of global plans and structural plans. Global plans involve: executive schemata, related to the overall control of the activity; and, a genre schema, derived from learning experiences and related to specific styles. Structural plans involve: a content processor, responsible for organizing incoming information instantiated by the genre schema according to rules of musical syntax, etc.; and, a music processor for translating the content into musical notation. An analysis of the subject verbalizations revealed a preference for planning activities which were related to lower-level content generation to the almost complete exclusion of long-term, higher-level planning. This preference can be explained in terms of processing requirements, as students at this stage in their musical development are most familiar with low-level knowledge structures and know little about how to combine those structures into musical units, or how to relate them to concepts of musical architecture.

REVISION

The model of composition revision presented in Hermann (1985), is based on the notion of revision being an internal reflection on interactions between representations of the intended sounds and representations of the actual sounds. A comparison is made between the intended sounds and the actual sounds. If a mismatch is perceived then a diagnosis is made as to what caused the mismatch. Finally, a tactic is chosen to operate on the original representation to try to alleviate the mismatch and the revision is completed. It was found that students who made use of the revising, recording, and playback capabilities of the computer music tool demonstrated a better use of decision-making strategies related to revision. In addition, those subjects who were given a special procedural facilitation in reference to the compare, diagnose, and operate sub-processes of the revision procedure demonstrated an increased ability to make changes for the better to the initial score.

THE COMPUTER AS A TOOL FOR TEACHING COMPOSITION

Of the three main musical activities in the music classroom – performing, listening, and composing – composition activities have not received the attention they deserve due to a lack of understanding of the composition process and what is learned through it as well as the lack of a resource for teaching it. In a manner similar to how the listening activity came to prominence with the development of a new tool, the gramophone, in the early 1900's, so another technological innovation, the computer, will enable the composition activity to come to prominence. Now a balance can be struck in the curriculum between performing, listening, and composing.

The microcomputer in the music education classroom offers a resource for encouraging composition activities. The sound editing capabilities of the computer music system can be used not only for an investigation of timbre, but also concepts of melody, rhythm, harmony, and texture. The music editing capabilities of the computer music system can be used in particular to facilitate the revision process in composition as students can get instant and accurate performance feedback without the need for renotating or performing the score themselves.

In connection with the above issues, is the need to examine effective methods of presenting the computer music tool to young learners and composers. Studies related to instructional design issues, such as interfaces, music editors, and effective screen displays should be undertaken in relation to fostering creative music activities in children. Such systems need to be sensitive to the wide range of individual differences which is sometimes found in the musical development and learning styles of students in a typical music classroom. The present study showed some difficulties arising from the design characteristics of the alphaSyntauri music editor for students with limited keyboard ability, for example.

SUMMARY

The results from this study show that students who had an opportunity to investigate in detail the acoustic concepts related to timbre through computer simulation acquired a deeper understanding of what the element of music known as timbre actually is. In addition, it was found that students using the computer music system in conjunction with their composition made more revisions to their scores than those who did not have access to the computer tool. Indications are that the music editor capabilities for recording, changing, and performing compositions facilitated the cognitive process of revision in composing. Results also show that the insertion of a procedural facilitation into the composing process helped to alleviate the problems of executive control in the revision process.

This interdisciplinary approach to music education research has shown how theories from cognitive science, music cognition, and computers in education interact to produce an understanding of the use of the computer as a tool to assist in the cognitive processes of composition. Finally, it is my belief that much can be learned through creative musical activities. Hopefully the computer music tool will be used as a valuable resource in placing music composition into the position it belongs in the music education curriculum alongside listening and performing.

REFERENCES

Hermann, B. (1985). Cognition and Music. Unpublished paper, The Ontario Institute for Studies in Education.

Hermann, B. (1985). *Computers and Composition in the Music Education Classroom: A Tool for Teaching the Cognitive Processes of Music Composition.* Unpublished master's thesis, University of Toronto.

Scardamalia, M. and Bereiter, C. (1983). The Development of Evaluative, Diagnostic, and Remedial Capabilities in Children's Composing. In M. Martlew (Ed.), *The Psychology of Written Language.* (pp. 67-95). New York: John Wiley and Sons, Ltd.

Upitis, R. (1982). *A Computer Assisted Instruction Approach to Music for Junior-Age Children: Using ALF for Teaching Music Composition.* Unpublished master's thesis, Queen's University.

Talking Books - Print Learning Materials Go Interactive

Michael Nitefor, President
Zvook Corporation
25 Adelaide Street, East, Suite 617
Toronto, Ontario M5C 1Y2

Thanks to the micro-chip, we can now produce books that talk, point, quiz and score. The sight-sound combination can help in both academic and on-the-job learning where immediate feedback is vital for effective reinforcement. There are opportunities for participation in pilot projects involving language study, skills training, special education, etc.

"That's correct", "Very good", "Examine this diagram", "Put on your thinking cap". That's what you'll hear along with hundreds of other assistive phrases when you use VOX.tm It's a newly developed learning aid that marries microelectronics to the book.[1]

The result: interactive books that can actually talk with learners. VOX not only talks to you while you're studying from a book, it can also point to text or graphics on a page. As well, VOX has the unique ability to operate a VCR, videodisc, or computer through its RS232 port to support learning taking place from a book.

The learner activates all these functions by simply scanning bar codes in a book.

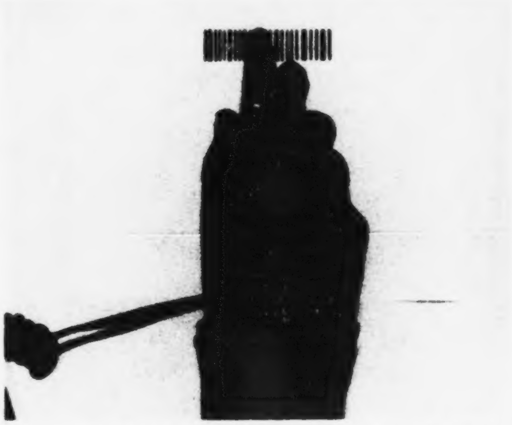


Figure 1. Control all VOX functions by simply scanning bar codes.

Open VOX just as you would a 3-ring binder. Inside, you see the learning material: a simple, paper bound book. The back cover of the binder houses the electronics which produce all of VOX's learning and testing functions. See Fig. 2.

Teachers, trainers and instructors who have seen VOX in action most often describe it as a learning facilitator. Perhaps its most attractive feature is that no computer literacy is required to either use or write 'courseware', i.e., books.

Leading Edge

VOX was designed from the ground up, in conjunction with the National Research Council of Canada, by Zvook Corporation, a leader in instructional technology development. The paramount goal was to facilitate learning in the real world of growing demands on teachers, expanding curriculum requirements (e.g., in skills training and special education) and the ever-looming spectre of tighter and tighter budgets.

Many industries have adopted technology to improve productivity. Now 'learning industry' professionals are starting to embrace the power and versatility of micro-chip technology. Zvook Corporation, the developer of VOX, back in 1981 noted a market niche for a learning aid product which would optimize the factors of low cost, portability, pedagogical soundness, easy integration into existing methods and procedures, ease of learning material preparation and, most importantly, measurable results in learner achievement across the full curriculum spectrum.

Areas of application

VOX versatility makes it at home in a wide variety of applications, settings and learner groups, enhancing and supplementing formal learning, and enriching personal learning in areas such as:

- learning to read, write and compute;
- traditional academic content and continuing liberal studies;
- acquiring second language capability;
- remedial, rehabilitative, and special education applications;
- professional training, upgrading, and updating;
- management training, upgrading, and updating; and,
- vocational training, retraining, upgrading, and updating.

What VOX does

In simplest terms VOX makes life easier for both teachers and students - and for taxpayers. How? By doing many of the things educational computers do now; providing rapid feedback, tracking, individualizing instruction, etc. VOX does all this and more using techniques and a medium already familiar to teach-

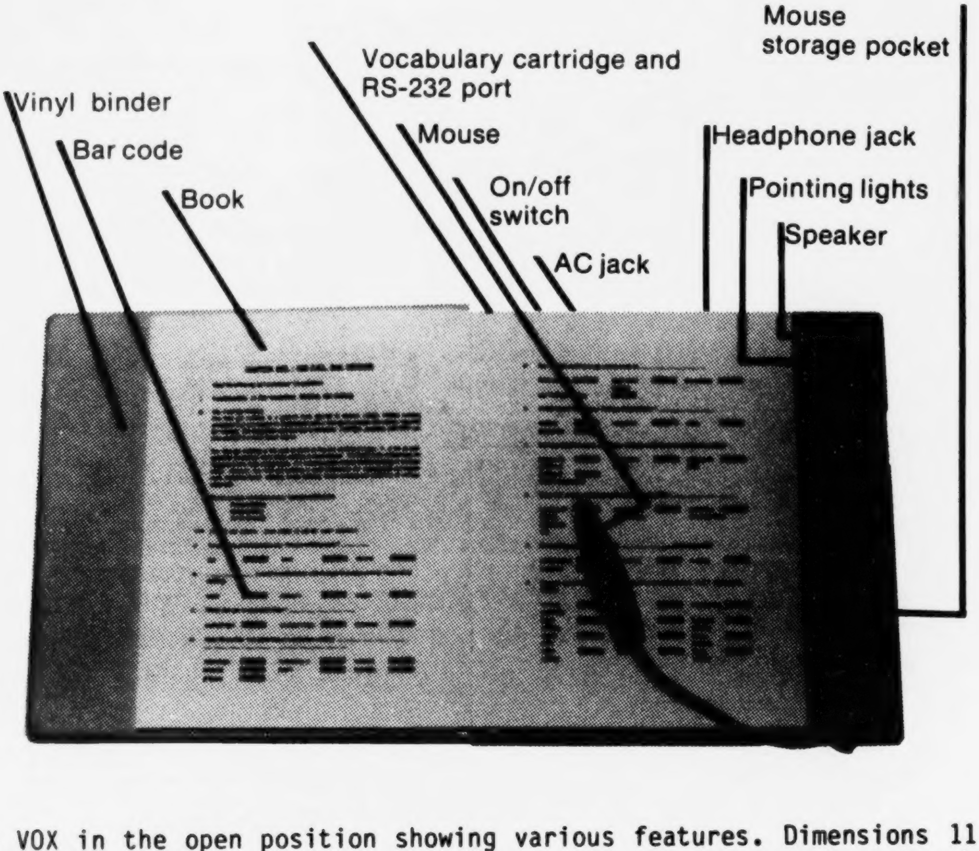

Mouse storage pocket
Vocabulary cartridge and RS-232 port
Vinyl binder
Bar code
Book
Mouse
On/off switch
AC jack
Headphone jack
Pointing lights
Speaker

Figure 2. VOX in the open position showing various features. Dimensions 11 1/2" x 11 1/2" in closed position. Weight, less than 2 pounds. Rechargeable batteries.

BEST COPY AVAILABLE

ers and students: the good old book. We believe the book is here to stay for a long time - and it certainly presents no VDT radiation hazard - real or imagined. Moreover, the cost is attractive to budget committees.

Functionally this is what VOX does. VOX talks to you. Just like a person - prompting, directing, correcting, encouraging and reassuring. All this in any language and with any intonation. A syllable, a word or entire sentences - each produced by a unique code.

Not only does VOX talk, it can make any imaginable sound. A bell, a properly running machine, musical tones, the sound of applause. Just scan a code.

VOX points to words and pictures in the book. It uses flashing lights instead of a finger. There is a row of lights along the right side of the book. By scanning a code you make one of them flash. Each light draws visual attention to an item on the page, or lets you know if you've chosen the correct answer to a question.

VOX helps you test yourself, verifying correct answers by pointing or oral feedback. Scan a code. If you get the answer wrong, VOX guides you towards the right one. Immediate feedback, for confidence building learning.

VOX analyzes your choices when you perform matching and ordering exercises. Your choice, activated by scanning a code, results in an oral prompt. Great for learning and recognizing patterns and sequences.

VOX can even give a test without supervision and lock in test scores. It can remember them, play them back and compare succeeding sets of data to chart the learner's progress. All under the control of the instructor.

VOX stores and plays back information. This allows you to store letters to make words, words to make sentences, or tones to make tunes.

VOX not only enhances the capabilities of the printed book, but interfaces books with existing audio-visual devices. Now, sophisticated equipment like VCR's, personal computers, monitors, videodisks or simulators, when interfaced with VOX, can be controlled by simple codes. A breakthrough in combining teaching media. Use more technology - without more complexity. For example: VOX is interfaced with a VCR. All VCR functions are then activated by scanning a set of control codes in the book. The codes appear in the book wherever access to the VCR is required to illustrate the text in live action. Your hands never have to touch the VCR controls. Just scan a code. Repeat any VCR segment simply by rescanning.

The VOX book

It's an exciting breakthrough in learning technique. A book which actually interacts with a learner and is used just as an ordinary book. Read it, look at pictures, scan codes as they appear in the text to activate various interactive support functions as 'programmed' in by the author. The only new skill required is learning how to scan codes. Most people learn in a matter of minutes. It's as easy as drawing a straight line with a pencil.

Developing VOX books is simple and cost-effective because, in many, if not most cases, you can use existing learning materials and simply adapt them by adding appropriate bar codes to elicit the assistive function you specify. Some existing materials may require a level of editing to improve their instructional design and to take full advantage of relevant learning theory and research. The addition of appropriate bar codes is a straightforward operation. In its simplest form, just attach self-adhesive code labels on a page. In a more automated process, any properly-equipped personal computer can accomplish this task along with its word processing capability. Using word processing, VOX learning material can be easily updated or revised and cost of reproduction is the same as for conventional paper-based material. No special paper or ink is needed. Xeroxes work just fine. The skills required to write VOX material are identical to those used in writing conventional materials - a factor teachers and writers are finding very appealing.

Past, present and future

The VOX learning system has now been developed to a stage where pilot projects are underway. Most recently Zvook Corporation entered into an agreement to supply VOX to the Public Service Commission of Canada for facilitating language training.[2] Other pilots are now being negotiated with several Community Colleges in Ontario focusing on skills training. As well, activity has recently been initiated in the private sector to use VOX as a training aid with a number of major companies representing a variety of industries.

Getting involved

The AEDS/ECOO Conference will be the forum for launching a special program for conducting pilot projects in educational jurisdictions outside of Ontario. (Ontario jurisdictions may also be eligible for this program if certain criteria are met.) In our continuing exploration of the potential of the VOX system we have been truly surprised by the degree of original work done by teachers. To build this momentum, and to add to the still small base of empirical data on the validity of VOX, we are inviting interested educators to work with Zvook to examine its potential in expediting educational and training delivery in their jurisdictions. Under contractual arrangements, we will supply a number of VOX units and assist in all aspects of learning material creation or adaptation, including formative evaluation, testing and final implementation. Inquiries should be directed in writing to Mr. Michael Nitefor, President, Zvook Corporation, 25 Adelaide Street East, Suite 617, Toronto, Ontario M5C 1Y2. No phone calls please.

VOX is a trademark of Zvook Corporation.

REFERENCES

1. Allan, Susan. "Portability Enhances VOX Reading Tutor", article in Technology section, Report on Business, Toronto *Globe and Mail*, Friday, November 1, 1983, p. B14.

2. Banks, B. "VOX Targeted at Gap Between Books and Micros" *Computer World Canada*, 1984. Vol1(2), p. 20.

Integrating Tool Software In The School And Classroom

John Burton
Champlain College
P.O. Box 670
Burlington, Vermont 05402

As computers become established in schools, the application of tool software (wordprocessors, databases, and spreadsheets etc.) to classroom objectives becomes a critical component of a comprehensive computer education program. This paper presents some ways that computer inservice activities can be structured to promote the use of application software in the classroom. Through examination of experiences with a mobile computer lab , the IBM Model Schools project and individual consulting this paper provides some insights about the effective use of application software in the school setting.

This Christmas a small robot walked into my life. This droid reminds me of the way that many fello educators seem to approach information technology in the schools. Although powered by only a, 1.5 volt battery, this "marvel of our age" can move forward, shake its head while pretending to read some little predefined plan (often held upside down at that !), babble incessantly, and charge off in a slightly different direction. If this "toy" (I use the term lightly) ever encounters resistance it immediately turns and proceeds in a new direction and of course after not too long it finds it has gone around in a proverbial circle. This must be why it has been marketed under the name - "DINGBOT"!

Many of our educational leaders seem to behave much like this android as they face the challenges involved with integrating information technology in their schools. Often understaffed and underfunded, our leaders talk about "implementation agendas" but proceed in a direction that doesn't have much to do with their plans. Seldom do reactions to resistance result in anything but a charge in another direction. Unless the productivity capability of the computer is integrated into the daily life of our schools, the lack of effective use may cause the computer to be viewed as a useless toy taking us around in circles.

Champlain College in Burlington Vermont has over 20 years of experience training business leaders the effective use of the computer as a productivity tool. More recently Champlain College established a Computer Resource and Training Center to assist schools in their efforts to introduce computers into their classrooms. As part of the efforts of the Resource Center, three approaches have begun to shed some light on both the process as well as content of implementing and integrating application software in schools.

The three approaches, although different in design have each proven successful in moving schools toward appropriate use of tool software. Using these three approaches as models of computer inservice delivery, educators can learn much about computer education as well as the nature of tool software

BEST COPY AVAILABLE

These three models can be labeled as infiltration, induction, and infusion. Although the models bear similar fruit they each have separate strengths and weaknesses with each approach offering aspects that prove promising for further development.

The INFILTRATION model involves working from within the existing talents and mechanisms of the school and thus building local resources leading toward an integrative capacity on the part of the school. Toward this end a consulting model can work quite well. During the past 18 months the Computer Resource and Training Center has had an ongoing relationship with several school districts in Vermont. The most fruitful relationship grew from the need on the part of one district wide committee to build a computer course of study for the teachers and by the teachers. The role of the consultant was one of mediator-educated listener. During the last year of intensive consultations this district to build a 3 tiered course of study beginning with introductory skills, proceeding through the major applications and into independent explorations. Lately the consultants role has evolved into an endorsement agent to validate the skills of prospective instructors from within the staff of the district and assisted the committee in evaluating the effectiveness of these inhouse courses. This model builds a strong and lasting inhouse capacity for continued computer development a reasonable costs. Another advantage of this approach is the use of instructors knowledgable about the realities of classroom life as well as school applications of tool software. Some of the difficulties inherent in this model are assuring quality and managing the support for such demanding courses.

In contrast to the infiltration approach, the INDUCTION model brings in an outside element to induce the school to integrate application software into all the various curriculum areas. During the 1983-84 school year Champlain College introduced the Computers in the Classroom program. This resource brings 1 instructor, 2 assistants and 25 microcomputers to a school for a two week period. The computers are set up in a free space in the building and classes are conducted for up to 125 students, 25 staff members, and 25 community members per week. The curriculum is developed by the local educators after conducting an extensive needs assessment and decision making process. Each school receives a 100 page manual and consultations to assist in the process of defining their curriculum. During the two weeks students are taught a wide range of computers applications with each session closely tied to specific classroom topics. As students from each teacher's classroom attend, the teacher participates as an assistant with each session planned in close coordination with the instructor and the local teachers. The after school teacher course focuses on integration issues and discussion emanating from the days instruction as well as further skill development in the use of the major tool software packages. The night courses offer parent and community members a chance to become familiar with productivity tools as well as the educational implications of their use. Often the school board or a district wide computer committee meets with the instructor for advice on implementing tool software in the classroom .

The inductive model's strength is due to the intensity of the experience and modeling of the

BEST COPY AVAILABLE

integration techniques. Many find the general interest and capability of the children, teachers and community increase to a point whereby it becomes easier to implement computer applications in the school. Teachers have remarked that the power of modeling the effective use of applications software can allow them to "see what to do" rather than simply describing it, as is done in many teacher education courses. Schools appreciate the low cost and reasonable demand this model calls for on the part of their local resources. The disadvantages include the grim reality that the equipment must go and the local resources are often so limited that it is difficult to get time with the computer. This can build what some refer to as "positive pressure" to purchase more equipment, and in many cases this has happened.

The INFUSION model of implementing applications software is the one adopted by the recent IBM model schools program. This hardware and software giveaway began last winter with an intensive training program for 2 resource people from each of 28 Professional Development Centers around the United States. Each team spent 3 weeks at a training institution (Univ. of South Florida or Bank Street College of Education) honing their applications software skills, developing teacher training skills and devising techniques for integrating tool software into the curriculum matters. During the month of July, 5 teachers from each of 5 local schools were chosen to attend a 4 week training program aimed at preparation for tool software use in the subject matter areas. The group of 25 teachers came from the various disciplines and received a wide range of experiences with word processors, databases, spreadsheets and telecommunications. During the fall of 1984 each school received 15 microcomputers from IBM as well as multiple copies of software appropriate for their students. During the school year the two inservice trainers have provided ongoing support to the teachers as they begin implementing the computer in their regular classroom activities.

This model has just begun implementation but appears to offer a substantial improvement over some of the other models because of the resources that have gone into assuring its success. Each school has the support from the Superintendent down to regular classroom teacher as well as the expertise and equipment to provide an intensive approach to integration. One of the weaknesses inherent in this model is the lack of a multiple year commitment. This kind of lengthy commitment may be required to continue supporting the curriculum development activities or many of the gains may not continue to grow and include more than the small core group of committed teachers.

Although these three models approach computer implementation from different directions each has taught some important things about not only affecting change in information technology and institutionalization of innovation but also about specific methods of using tool software in the school setting. Each tool software category has many facets and unique qualities associated with it. We have much to learn about each of these and the ways in which children react/interact with them. Some observations about these software attributes are summarized below:

BEST COPY AVAILABLE

Word Processors-

Word processors allow children to explore the world of written expression. In and of themselves word processors are in open ended teaching tool, allowing the teacher to structure lessons and interaction in a variety of ways. They can be used to highlight concepts, model proper communication behavior, share styles and attitudes, view similar information in different ways and generally encourage people to learn through doing and interacting rather than just viewing from the sidelines. Word processors can be used in highly creative ways to teach a multitude of concepts ranging from poetry to outlining. Both students and teachers need to learn to open their creative energies allowing them to explore the fantastically pliant media called word processors.

Data Base Managers-

Data base managers hold the key to information technology in our society and schools. When children and adults work with a database manager they come head to head with information and how to best structure it for specific ends. Databases bring to the fore diverse subjects in unique ways. Todays Educators can structure Databases in such a way that they encourage certain types of conclusions and observations, many in unexpected ways. Discussion of hypothesis testing and revision as well as assistance to create solutions, can be just a few of the many objective of database use.

Spreadsheets-

Spreadsheets give mathematics and science students a simple "breadboard" allowing for the testing of patterns and observation of outcomes. Spreadsheets provide students who need repeated exposure to patterns, the ability to interact endlessly until the eureka experience crystallizes. When linked to graphics software they provide the all important tie to the right side of the brain that many students require for maximizing their learning. The structure of the Spreadsheet can be as simple or as difficult, as open or as closed as the lesson or student requires. Spreadsheets can remove the painful boredom from repetitive calculations and unlock new vistas and possibilities.

This paper outlines some initial steps and observations toward integration of tool software in our schools. Much has been learned with as well as about informational technology in education but a great deal is yet to be discovered. Let us hope that we can continue to break out of our old mold and become something more than a Dingbot in our training techniques developing information technology in our schools.

Any questions or comments can be send to the author at Champlain College P.O. Box 670 Burlington, Vermont 05402 or through e-mail on compuserve # 70014,2106 .

BEST COPY AVAILABLE

Computing Tools In The Classroom: Through The Curriculum And Beyond

Henry F. Olds, Jr.
17 Maple Terrace #3
Belmont, MA 02178

Many trends are currently combining to make more common in classrooms the use of general purpose computer tools for supporting and extending the curriculum. Teachers and administrators are learning the many applications for these tools. Publishers are creating applications' packages to aide teachers in their use. And new tools designed for the student are offering features that significantly expand educational applications. As use of these tools grows, the curriculum and the nature of education will change.

Only a few years ago, the prevailing view about using computers in education was that they would serve, like the textbooks they might partially replace, as valuable adjuncts to the teaching process. Software would consist primarily of a series of tutorials and associated record keeping utilities for tracking student performance. Today that view is giving way to another that grows out of a more complete understanding of the potential of computers for providing teachers and students with a limited array of very powerful general-purpose tools for conducting most significant educational tasks.

Let's take a brief look into the near future of about two years from now. Linda, a seventh grader, stands in front of her locker checking her small storage case of computer disks. She verifies that the case contains all the tools she will need today: a single computer program (on one computer disk) that integrates word processing, file management, numerical calculation, illustration and graphic display; several file disks containing her own special tool applications (templates) that she has created for use in different subject areas; and several data disks containing recently completed assignments and work in progress.

In every class Linda goes to during the day, she will make use of some portions of her computer program. In social studies class, she will work with some friends to design a file for storing the information they need to carry out a comparative study of the major river basins of the world. The group will first enter the outline for their file in the word processor and print out copies for each person to take home. Tomorrow they will make any needed revisions, create the file format and then begin to enter some of the data they have already collected.

In math class, Linda will work with another group of her classmates to solve a set of interrelated problems that grew out of a simulation of the economy of a small island community, all of which require working with numerical data that has been provided to them in several data files. They will need to find the numbers they need by conducting a series of searches through the data with their file manager, perform a series of calculations on these numbers using the calculator along with some templates they have already designed for such standard routines as computing means, create several charts and graphs to present their results by using the graphing utility and write up a defense of their conclulsions on the word processor.

In English class, Linda will print out her data file of figurative uses of language which she has been maintaining for several months. Following her teacher's assignment, she will then draft a description of a person in which she will try to use as many of these items as possible.

In science class, Linda will will use the illustrator to draw several different kinds of leaves and label them. Later in the week, she will enter into her data file her notes about the trees they came from.

In short, in every subject, and for a wide variety of purposes, Linda will use her computer tools regularly. At no time will she have taken any course in how to use them because beginning to learn to use them is simple and straightforward. But as she has more and more experience with them and attempts to use them in more ways, she learns increasingly more about functions they can perform and advanced techniques for achieving mastery of their use. There is much sharing of knowledge and skills about tool use among the students. A common topic of conversation among Linda's classmates is any new technique or template developed by a classmate for solving a particular type of problem.

Linda still uses tutorials, games, and puzzles occasionally for learning particular things she knows she needs help with. She uses simulations even more, frequently to set a context within which she then applies her tool kit to help her understand a problem, analyze it and solve it.

Many schools are already well on the road toward this future reality. At this stage, tool use is largely separated and departmentalized, following the standard division of subject matter areas. Word processing is becoming acceptable in English classes, file management in social studies, calculating and graphing in math, illustration in art, etc. It's a first and important step. Not many schools yet make use of these tools across subject areas, but there is growing recognition that it is not only possible but desirable. In most districts, the availability of computers in all classrooms in the school is still limited, and the software programs are just now being purchased in quantitites sufficient to serve all students and teachers.

Major educational publishers are beginning to help further the process of tool adoption and use by publishing materials that relate the tools to their standard textbooks through a series of templates designed to fit closely with the topics studied in the book. Thus teachers can comfortably begin to use the computer with their students as an adjunct activity to further reinforce the knowledge and skills they normally teach. One example from a set of file management templates designed to accompany a fourth grade social studies series gives the student a file which requests twenty items of information for every state in the country. The information for the Northeast is already filled in so the students can see how it's done and have a chance to develop some initial skills in searching, sorting and printing. Then, as the class moves through the book and the rest of the regions of the country, the students must fill in the information for every other state. Many activities are possible with this data base as it grows, and a short teacher's manual, which includes worksheets for the students, provides a number of suggesions. New materials like this, that effectively link the computer as a tool with curriculum and build both computer skills and enhanced subject matter understanding, will create a good foundation for moving education toward kind of program available in Linda's school.

It is only recently that program developers have turned their attention to creating software tools of high quality that are designed specifically with the needs of teachers and students in mind. Recent word processors have many useful educational features, including windows, color highlighting and simplified methods of moving text about. Several writing programs now permit text and illustrations to be combined, opening the door for all students to experiment with visual as well as verbal communication. File management programs provide enough features to do most important educational tasks with speed of execution and simplified procedures.

There is a strange lack of programs that permit the computer to function like a simple programmable hand calculator, but more will surely be avilable soon. As more of these tools become available, as the features they offer multiply and as their ease of use improves, they will be increasingly welcomed into classrooms.

Gradually, because students like Linda will learn to use these tools to improve their own learning and to extend the range of their minds in even more ways than we can possibly predict, the nature of what happens in schools will change. Perhaps the traditional course of study that Linda still follows will give way to programs of study that more clearly than ever focus the attention of education on nurturing the power of the mind, through both its internal resources and its external tools.

Competency Based Computer Education

Gene Sullivan
University of Michigan-Flint
Flint, Michigan 48503

Computer education needs to be defined with more precision than has been the case in the recent past. Colleges, universities and school districts can more clearly communicate what is meant by computer education and computer literacy if they design courses and prescribe curricula in terms of performance based computer compentencies. Computer literacy should be viewed as a process. It is the process of learning basic terminology and becoming competent with specific computer applications.

STATEMENT OF THE PROBLEM

The phrase computer literacy has been used in many ways in the literature related to computer education. Nearly every article I read about computer education uses the term, yet few are precise about what computer literacy means. According to Pantiel and Petersen in their book Kids, Teachers, and Computers, it has "...become an educational buzz word, taking on a somewhat indiscriminate range of meanings over the last few years. Unfortunately, it has been used to define everything from a simple demonstration of how to turn on the machine and type on the keyboard to having a fundamental knowledge of how to program the computer." (Pantiel and Petersen, 1984) One recent nationwide survey polled the 50 states with questions sent to state departments of education in regard to "computer literacy" with no definition of the term. (Abernathy and Pettibone, 1984) The results of such surveys without specific meaning for the term are of limited usefulness. In an article in Currents, the author says, "There seems to be unanimous agreement that computer literacy should be an important part of every curriculum from kindergarten through college and that this should involve not only teaching about computers but teaching how to program." (Tinker, 1983) There is little debate about the first part of this statement. However, a casual reading of a few articles, reports or surveys would lead to the conclusion that there is little agreement about what should be involved in the computer education curriculum. The ubiquitous nature of the term computer literacy allows it to be used to mean everything without meaning anything.

Those working on the National Assessment of Educational Progress project had to confront the issue head on. NAEP plans to conduct the first evaluation of computer competency in the spring of 1986. It was noted that trying to measure computer competency was like trying to hit a moving target. It was concluded that the assessment should include: (a) using computer based tools (applications), (b) designing computer based tools (programming in BASIC, PASCAL or LOGO), and (c) understanding how computers shape society (knowledge and attitudes about computers). A question that remains to be answered is whether a computer assessment can be adequately accomplished with a paper-and-pencil test -- without a computer-performance test.

To complicate the definition problem a little more, Irwin Hoffman in an editorial in Electronic Education, contends that computer literacy in schools today is nothing more than a stopgap measure and that when the metamorphosis of this curriculum is complete it will center around word processing and complex databases cutting across the curriculum to meet the needs of the Information Age. (Hoffman, 1984) If computer literacy includes programming, applications, computer ethics, and computer awareness, what specifically do these terms mean?

THE NEED FOR CLARITY

Clarity and precision in terms related to computer education are essential for clear communication between the many parties normally a part of the planning and implementation of computer education programs. Administrators, board members, teachers and parents need to be clear about the meaning of computer education as they plan for their school system.

It is equally important that computer education evolve from an ambiguous to a focused state to make the maximum use of limited human and material resources. Waste from lack of direction and trial and error approaches is costly and unnecessary. School days and school budgets are limited and require clear cut plans for computer related expenditures.

COMPUTER LITERACY AND COMPUTER COMPETENCE

It is the contention of this paper that computer literacy should be seen in terms of computer competence. Competency based computer education should identify specific knowledge, skills and attitudes expected at various levels of computer literacy. While the term competence is not limited exclusively to behavior, it is often associated with performance objectives which can be clearly defined, observed and measured.

A second contention is that computer literacy is the result of becoming more and more knowledgeable about and skilled with the computer. As individuals gain competence with the computer they become computer literate in clearly specified areas. It stands to reason that computer literacy isn't a state of being but rather the process of becoming.

COMPETENCY BASED SOFTWARE EDUCATION: AN EXAMPLE

One example of a competency based computer education course for teachers based on the premise that computer literacy is a process integrated with competence in computer related knowledge and skills is a course offered at The University of Michigan-Flint. At the beginning of the course, offered for preservice and inservice teachers, class members were given a pre-test to determine their level of computer knowledge and skill in selected areas. In addition to basic computer terminology, assessed with a 50 item instrument, participants were given seven problems to test their skills in computer operation, word processing, data base management, spreadsheets, and LOGO. Six written evaluations of educational software were required from each participant.

Expected competencies: By the end of the course the participants were expected to be able to:

1. Start up or "boot" an Apple lle and IBM PC.
2. Operate two word processing programs (Bank Street Writer and PFS Write).
3. Operate a data base management program (PFS File).
4. Operate a spreadsheet program (Lotus 123).
5. Write simple procedures using LOGO.
6. Correctly identify selected computer terminology.
7. Evaluate computer courseware.

The format of the fifteen three-hour sessions included an introductory lecture-demonstration segment followed by a computer lab segment. Extra out-of-class lab time was necessary to complete the periodic competency check assignments. The classroom was equipped with eight IBM PC's and one Apple IIe. Normally two students worked at a computer and this proved to be a good arrangement because of the peer teaching and supportive interaction that took place.

RESULTS

The results of the pre- and post-tests are given in Figure 1. The class averaged 18.25% correct on the performance pre-test and 91.5% correct on the post-test for an average gain of 73.25%. The gain was less dramatic for the written segment of the test, where the average was 55% correct on the pre-test and 96% correct on the post-test for an average gain of 41%.

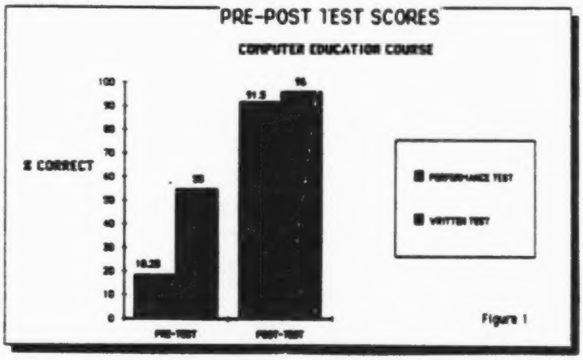


Figure 1

It is possible that the pre-post test differences for the written segment were less pronounced because of the general knowledge about computers students had prior to the class. It is also possible to guess on some paper-and-pencil items. The performance problems used virtually eliminated the possibility of guessing. The periodic competency check assignments given during the course provided valuable feedback to the students and the instructors concerning individual progress in the seven tested areas.

The overall format of the course proved quite satisfactory. Clearly defined goals allow for easy modification of the course if desired.

Student evaluations of the course suggested that there were important attitudinal changes because of the course. The next step is to design appropriate instruments to measure growth in this area.

REFERENCES

Abernathy, S.M., and Pettibone, T.J. "Computer Literacy and Certification: What States Are Doing", T.H.E. Journal, 1984, Vol. 12(4), p. 117.

Hoffman, I. "When Computer Literacy Metamorphosis is Complete...", Electronic Learning, 1984, Vol.4 (3), p. 18.

Pantiel, M., and Petersen, B. Kids, Teachers, and Computers, Englewood Cliffs, New Jersey: Prentice-Hall, 1984, p. 11.

Tinker, B. "Logo's Limits: Or Which Language Should We Teach?", Currents, 1984, Vol.2(2), p. 17.

BEST COPY AVAILABLE

Spreadsheets And Decision Support Systems: Tools And Environments For Teaching And Learning

Gary Marchionini
University of Maryland, College Park
4121C Hornbake
College Park, MD 20742

Electronic spreadsheet programs are being used in classrooms for a variety of reasons. They are studied as electronic tools in classes aimed at computer literacy or problem solving and decision making. They are also used as environments for the study of traditional content like mathematics, social studies and physics and as laboratories for research on human cognition. Decision support systems are becoming available for microcomputers and promise to be used in similar ways. The development of templates for these packages is compared to programming in a formal language and suggestions for teaching these packages are made.

The electronic spreadsheet program is an example of a generic application of the computer which has found multiple uses in many areas, including education. Beyond the obvious uses of spreadsheets (SS) for administrative planning and scheduling, SS are finding their way into classrooms as objects of study in their own right (as electronic tools) and as environments for the study of traditional content. Parallel developments are occurring with word processing and database management software packages. A new genre of microcomputer software termed decision support systems (DSS) is emerging and offers great potential for classroom use. The purpose of this paper is to examine ways SS and eventually DSS can be used in learning and to suggest ways for teaching the use of these application packages.

An electronic spreadsheet is a rectangular array of cells, each of which can take a lable, a value, or a formula relating any of the cells. The row by column format is a simple yet powerful structure for organizing information and representing it visually. The computer's ability to perform rapid calculations together with the capabilty of relating any cells allows users to test "what if" hypotheses by changing any cell value and watching all related cells be instantly updated. These properties allow users to deal with complex, abstract situations at a visual level. Kay (1984) summarizes this notion by stating: "The easy transition from the visual metaphor to the symbolic value rule brings the full power of abstract models to bear almost without notice (p. 57)."

Whereas SS are a product of the microcomputer which have migrated to mainframe systems, DSS are products of large systems which are just beginning to migrate down to microcomputers. Davis (1984) distinguishes between management information systems (MIS), DSS, and knowledge based systems (KBS). MIS manage huge amount of data, DSS allow manipulation of data in a structured way—the questions or alternative decisions must be known when the system is designed, and KBS assimilate new data in a manner which allows free form questions to be asked. Typical mainframe DSS are available for specialized situations and linked to extensive databases relevant to the given problem domain. Early microcomptuer-based DSS are more like SS in that a shell or structure for organizing information is given and the user determines what information is to be entered and how various cells should be weighted with respect to other cells. One DSS especially designed for microcomputer use provides a hierarchical (tree) structure for information organization rather than a rectangular array. Both SS and DSS provide command languages which can be very complex. Both require templates (structures designed for a particular task) to be developed. Once a template is completed, it can be used routinely by changing key values any time that task needs to be done. SS and eventually DSS can be studied to accomplish various learning goals. Two views of these programs are as tools and as environments. Examples from both of these viewpoints are described below.

Spreadsheet programs can themselves be objects of study. Just as programming is included in computer literacy courses so that students can apply the computer in intelligent and varied ways, SS programs can also be studied as part of becoming computer literate. SS and DSS programs can also be studied as generic problem solving tools. They provide organizing structures and facilitate systematic strategies for manipulating the data contained in the structures. In courses where problem solving or decision making are the objects of study, the focus is on the problem or decision and the programs act as electronic tools for attacting and solving the problem. The tools are studied to illustrate common processes for problem solving.

Just as the computer itself has become a metaphor for human information processing and an environment for studying human cognition, application programs can serve as environments for studying cognition. Arrays and hierarchies are powerful cognitive structures which extend human information processing through organizing (chunking) data. SS and DSS programs combine the computer's ability to perform rapid calculations and logical comparisons with these natural organizing structures to provide useful tools for thought and perhaps serve as explicit models of cognition. Studying peoples' use and design of various templates may lead to better theories of human cognition.

Of more immediate concern to classroom teachers is the use of SS or DSS as environments for teaching traditional content. The mathematics classroom is a natural place where students can apply their knowledge of formulas and mathematical relations in a SS environment. Mathematics teachers can use SS to introduce and develop a variety of mathematics concepts and students can explore mathematical algorithms by easily simulating their behavior via the SS. Arganbright (1984) presents several examples of using a SS for various high school mathematics topics. Social studies teachers can use SS to develop economic and political concepts. For example, Martorella (1984) presents a strategy for exploring presidential election projections via a SS. Physics principles which require knowledge of differential equations can be studied through the use of SS by students with only algebra experience. A physics course using SS is under development at the University of Maryland. In a business curriculum, accounting principles can be made the focus of attention rather than the tedium of calculation necessary with a paper and pencil spreadsheet. Clearly, as instructors gain more experience with SS and soon DSS, they will find additional uses for them in teaching traditional content.

Regardless of the reasons for using SS or DSS in learning, students must learn to use the programs in efficient and effective ways. Using and developing templates has much in common with using and developing programs in high level languages. Like programming languages, all SS or DSS offer a command set and a set of rules for using the commands. Unlike programming languages, SS and DSS offer a single data structure--the array or in some packages a tree. This limited way of representing data makes the "programming" easier, but less general. SS and DSS are also more special-purposed than formal programming languages since they are designed to be dynamic and demand interaction. Although these application packages are special purpose languages, many of the principles of teaching and learning programming apply.

A dilemma facing all teachers is the fact that experts use top-down procedures for doing programming, but skills and concepts must be built up from simple to complex. A common way of meeting this challenge is to focus on the general concepts of programming or template development by using a developmental sequence of activities to build and reinforce these concepts. Novices should be given simple example templates to use so they begin acquiring the command skills for a particular program. They can then be required to document (add labels, write descriptions) or modify existing templates. Simple problems can then be attacked from a design view, and templates created from scratch. Principles of modularization, documentation, testing, and debugging can be developed as more complex problems are considered. The level of detail included and amount of time spent studying a package depend on the eventual goals for the course. For all purposes however, the focus should be on the concepts of template design and dynamic use rather than the idosyncrasies of a particular pacakage.

Questions remain about learning to use SS and DSS, e.g., When should novices be introduced to forward references and self references? How can a sequence of keystrokes be best represented for sharing with others? Are sample "lists" (the labels and formulas) and "runs" (screen dump of a template with values entered) useful when studying such dynamic packages or are written analyses and descriptions more valuable? Research on these and other questions must be done. One thing seems certain however, SS and soon DSS will become increasingly available in educational settings and extend the uses of computers in education.

REFERENCES

Arganbright, Deane. Mathematical applications of an electronic spreadsheet, In Computers in mathematics education 1984 Yearbook of the National Council of Teachers of Mathematics (V. Hansen & M. Zweng, eds.). Reston, VA: NCTM, 1984.

Davis, Michael. Anatomy of decision support. Datamation, June 15, 1984, p. 201-8.

Kay, Alan. Computer software. Scientific American, September 1984, p. 53-9.

Martorella, Peter. Calling winners: using spreadsheets to project election day tallies. Electronic Learning, September 1984, p. 56-7.

Oracle (Consultant) Science Fair Research Project

Robert Byrnes
Smithfield Middle School
175 Mount Olive Drive
Rexdale, Ontario M9V 2E3

This science fair research project uses the Oracle (Consultant) data base to increase the efficacy of the library by providing a broad research information base for students. The program can be easily developed using parent volunteers and maintaining up-to-date files is simple and fast.

Smithfield Middle School, in Etobicoke, has had a science fair for the past ten years. The majority of entries are from grade eight supplemented by entries from grades six and seven. As part of the science curriculum all grade eight students research a topic for science. Our resource centre has over 150 students doing in-depth research on specific science topics.

When students come to the resource centre in small numbers we are able to give assistance and direction in choosing a variety of resources and to check their appropriateness. To assist the large number of students in locating a variety of appropriate resources dealing with their topic, I implemented the data base program Oracle (Consultant) using our existing Commodore equipment. Students found that their topics seldom matched titles in the subject card catalogue. A student researching a specific aspect of plants would initially look under plants in the subject card catalogue. The following are the titles that would be found:

- Plant Propagation
- The Mystery of Migration
- Plants That Eat Animals
- Finding Out About Green Plants
- Green Thumb In The Classroom
- Growing Things Indoors
- Growth From Seeds
- How Plants Are Made
- How Plants Grow
- Economic Plants
- Identifying Parasitic Plants
- The Kinds Of Plants
- Plant Hobbies: A Beginners Book of Gardening Projects
- Plant From Sea To Sea
- Non Flowering Plants
- Plants and Seeds Study Prints
- Insect Eating Plants
- Colour In Plants And Flowers
- Structure Of Plants
- Plants That Move
- Bits That Grow Big
- Plants Experiments
- Men's Useful Plants
- My Plant
- Collection And Preservation
- Edible
- Soilless Culture: Understanding Hydroponics
- Hydroponics
- Photosynthesis: Light: Colour and Life For the World
- Reproduction
- Tropisms: Phototropisms, Geotropism, Thigmotropism
- Uses

Often very general book titles give few clues to the specific information between the covers Students would have to peruse many books in order to find appropriate materials and a great deal of students' time would be taken in locating good material. We found that the majority of students would not go beyond two sources and these were primarily print material from non-fiction books. Students often become frustrated by spending an inappropriate amount of time selecting material.

Our resource centre provides a wide variety of materials using different media for research. We have non-fiction books, encyclopedias, National Geographics and other periodicals, study prints, slides, kits, filmstrips, vertical files and video tapes in addition to an audio visual catalogue supplied by our board.

Using the Partners approach in the resource centre, the science teacher and I pulled the science related non-fiction books from the shelves. The books were separated into two categories: those appropriate for science fair research and those not appropriate for this purpose. I then marked each selected research book with a red SC on the pocket and circulation card to assist us in future identification.

The science teacher and I identified certain research topics that consistently appeared over the years. Plants was the first topic we prepared for the data base.

A parent volunteer, who was a former science teacher, pulled out all of the non-fiction books dealing with plants that were previously identified. She used the index and table of contents to locate data that would be suitable for researching the following headings and sub-headings;

- HYDROPONICS
    - Nutrient Solution

PHOTOSYNTHESIS

- Food
- Carbon Dioxide
- Oxygen
- Chlorophyll
- Chloroplasts

REPRODUCTION

- Budding
- Cutting
- Cross-pollination
- Grafting
- Spores
- Seeds
- Runners

TROPISMS

- Geotropism
- Phototropism
- Thigmotropism

USES

- Food
- Clothing
- Shelter
- Medicine
- Spices
- Drugs
- Drinks
- Oils and Fats
- Manufactured Products

As satisfactory material was identified the following format was designed.

TOPIC: PHOTOSYNTHESIS
SAMPLE FILE
Source: Non-Fiction
Title: Leaves
Author: Caulfield, Peggy
Dewey #: 581
Letters: CAU
Volume:
Pages: 26
Cat #:
Comments:

Encyclopedias, National Geographics and other periodicals and the audio visual materials from Smithfield and our board catalogue were searched and satisfactory data was entered into files. Pertinent articles from older periodicals and encyclopedias were photocopied and placed in our vertical file.

With the implementation of the Oracle (Consultant) program we now have 81 specific references for students to use to assist them in researching plant related topics. (see appended chart)

Not only will this save students considerable time but they should have a broader knowledge base on which to develop their projects. The majority of their efforts can now be channelled into retrieving and applying knowledge.

The end result must be weighed against the time taken to implement a project such as this. As with all new programs we made errors and wasted some of our time doing unnecessary tasks. We now have streamlined our procedures. A realistic breakdown of the time needed to complete the task is outlined as follows:

TASK	TIME
Pull Books	1 hour
Select material from books	8 hours
Selecting materials from other sources	5 hours
Input of data	2 hours
Total	16 hours

Because our science fair is yearly, each topic can be built upon every year. As new materials becomes available a new file is entered into the record. Adding and deleting files takes seconds and is not a problem.

We are continuing to develop records for other topics and to date the following have been completed: Light, Electricity, Crystals, Energy, Magnetism and Machines.

We do not expect to be able to service all students or all topics with this computer program. It does, however, allow me time to assist the smaller unserviced groups locate satisfactory materials. I am able to assist all students with the skills required to extract and interpret the knowledge required for their science fair hypothesis.

Statistics: Oracle Data Base For Plants

Heading	Non Fiction	Encycl.	Kt F.S.	Movie Cat.	Vert. File	Periodicals	Total Sources
Hydroponics	2	1	-	-	-	-	3
Photosynthesis	9	6	2	1	-	-	18
Reproduction	12	15	1	2	-	-	30
Tropisms	-	3	-	1	-	-	4
-Geotropism	4	1	-	1	-	-	6
-Phototropism	6	4	-	1	1	-	12
-Thigmotropism	1	-	-	-	-	-	1
Uses	2	3	-	1	-	1	7
TOTALS	36	33	3	7	1	1	81

F.S. = Filmstrip

KT = Kit

Databases, Bulletin Boards and Information Retrieval: Promises and Pitfalls (Or Should I Buy A Modem?)

Susan E. Moont
Computers and Learning
55 Harbour Square Apartment 2013
Toronto, Ontario M5J 2L1

The development of the computer is the result of the need for society to collect, organize and distribute information. In fact, cave paintings could be considered to be the first databases. This urge to collect and manipulate data will continually effect the development of the computer. However, it should also have an impact on curriculum. The session will look at databases, bulletin boards, and information retrieval as part of classroom activity.

A database is a collection of mostly textual information stored on a computer. The amount and variety of text stored relates directly to the size of the computer. This sounds similar to a word processor, but there is a vast difference. In a word processor text is stored in sentences, paragraphs, chapters, and retrieved by a document name. In a database, information is stored by single words, phrases, or extremely structured text that must fit certain rules. Only very large, specialized databases can contain normal literary text.

Every person has been associated with databases for a large percentage of their life, including tax files, employee files, driver records, insurance files, credit card files, and department store accounts.

Many administrators rapidly recognized the potential of database application. There has been slower recognition of the value of database use in the classroom. However, database use is an excellent way of having many students utilize a single computer that may only be available periodically.

When students use a database application they learn many other skills as well as those related to the computer. There is a large research component attached to the development of any database. The students also see the need for extensive prior planning and problem solving. Using a database also increases students' exploration of terminology related to the topic they are studying.

A number of examples will qualify the benefits of database use. These are not my ideas, they come from books, films, and innovative teachers that I know. All of the ideas have been tried and they all work.

Beverly Hunter's book "My Students Use Computers", outlines how to use a database for the study of states, and for the study of planets. One topic that is ideal for database use is nutrition. Once students have compiled information on a number of foods they can do an effective analysis of their caloric and vitamin intake. One database topic that is great fun is rock groups. Creating this database can lead to the development of all kinds of communication skills.

In Australia, students studying history use a database called "The First Fleet". This contains the records of the convicts who were sent to Tasmania. Students gain first hand experience at historical analysis. They are able to discover for themselves what kind of crimes led to the long trip to Australia. They can look at the ages of the convicts and the places that they came from.

Databases are a valuable way of studying heritage or multicultural studies. The journal "The Computing Teacher" had an excellent article on an Indian study in the November 1984 issue. Science and geography abound with material suitable for use. Birds, rocks, animals, elements, and chemicals are well suited to classification activities. Demographic information of any kind also lends itself to database study.

Electronic Bulletin Boards

Most of us have been exposed to a variety of Bulletin Boards, at school, at work, and in the supermarket. Bulletin Boards contain rules and regulations, buy and sell ads, news of upcoming events, announcements of birthdays, awards, and graduations, and the occassional joke and graffiti. Electronic bulletin boards (bbs) provide the same kind of service, without the physical need to go to the bulletin board.

In order to participate in a bbs you need:

i) a computer
ii) a modem
iii) a phone line
iv) a communications program
v) access to a bulletin board (including a phone number and password)

There are three kinds of modems. A modem (modulator /demodulator) is a device that converts a computer signal into a phone signal and vice-versa. The type of modem that has been around the longest is called an accoustic coupler. It is a box with two rubber suction rings on top into which the phone receiver fits.

I use two terms of my own to describe the other two modems. The first is the inboard modem. It sits inside the computer and all that can be seen is a telephone cord with a regular telephone jack on the end. The other is the outboard modem which is a box that sits next to the computer. It is connected to the computer by means of a serial interface card, and it also has a length of telephone cable with a jack on the end. The most common example of the outboard modem is the "Hayes Smartmodem".

Modems are like LP records. They have a certain speed (of transmission and reception). There are two speeds common to modems used with microcomputers, 300 baud and 1200 baud. The modem that you use must have the same speed as the modem on the computer that you are communicationg with. Some modems, such as the Hayes Smartmodem work at both 300 and 1200 baud.

The heart of the bulletin board is a dedicated computer. In other words you can't do anything else with it. The central computer is the depository where everybody leaves their messages. Most bulletin boards are divided up into several different categories according to the interests of the people who use that bulletin board. Anybody can leave a message or respond to a message. As well there are usually mail (email - electronic mail) facilities where individuals who belong to the bulletin board can communicate privately, and a feedback option so that messages can be left for the sysop (system operator).

Knowledge of the existence of bbs passes by word of mouth and user group magazines. Most bulletin boards contain a list of other local bulletin boards. Once you gain access to one bb, then you find out about many more. There are hundreds of bulletin boards across Canada and the United States.

The biggest potential for using electronic communications of the bbs type is for the physically disabled. When two people communicate through a computer system, it is simply two people communicating. Most communication barriers just simply disappear as long as a reasonable resemblance to grammar is used. The use of electronic communications can often be a powerful motivator to make students improve the quality of their writing.

Information Retrieval

There are now many extremely large scale databases and electronic bulletin boards available to the public. "The Source" and "Compuserve" are examples of large bbs. "BRS After Dark" is an example of a large scale data base in the States. "Infoglobe" is an example of a large scale database in Toronto (Canada). M.David Stone 's book "Getting On-Line" has a good section on large scale information utilities.

Students have the possibility of accessing kinds of information for research that they haven't been able to use before but should have. For example, "Infoglobe" contains the entire text of "The Globe and Mail" newspaper since 1977. Students can call up current information that is not available in any other form.

Information stored in an information utility or large scale commercial database is non diminishing. In a library, once all the books on a subject have been borrowed there are no more resources until the books are returned. With a database it doesn't matter how many people have pulled out information, the original data is still there.

However, there is a cost associated with all of this activity. Charges are billed for the time that you are accessing the host computer. In addition there may be long distance phone charges.

DATABASE BIBLIOGRAPHY

INTRODUCING DATABASES

Grosswirth, M., Data bases to guide the travelling gourmet. Personal Computing, July, 1983. 120-123 / 125

Special Report: Information. Popular Computing, June 1984
Cronin, B. If information is power, ... 104
Miller, M. J. & Goley IV, G. F. Making sense of database software. 106-109/194-197
Zacks, N. Database managers come home. 117-122

Wilburn, G. & Wilburn, M., Demystifying databases. Infoage, April 1984. 7-8/46-47

DATABASES IN THE CLASSROOM

Downes, T., Handling information in the classroom - a novel approach to the use of data bases in years 4-9. Computers and Education: Dreams and Reality. Proceedings of the Second Australian Computer Education Conference, Sydney, September, 1984. 105-109

Dunfey, J., Using a database in an English classroom. The Computing Teacher, November, 1984. 26-27

Hunter, B., My students use computers: Learning activities for computer literacy. Virginia: Reston (Prentice-Hall), 1983.

Kalkstein, P., Computers on the playing field. Electronic Learning, Vol 3, #2, October, 1983. 89-92

Patrick, S., Using a database. Exploring English with a microcomputer, London: Council for Educational Technology, 1983.

Pon, K., Databasing in the elementary (and secondary) classroom. The Computing Teacher, November, 1984. 28-30

Wills, S., Doodle design debug: Process vs content issues in classroom computing. Computers and Education: Dreams and Reality. Proceedings of the Second Australian Computer Education Conference, Sydney, September, 1984. 90-96

ACCESS TO LARGE DATABASES

Lisanti, S., The on-line search: Accessing the world's electronic brain. BYTE, December, 1984. 215-230

Stone, M. D., Getting on-line: A guide to accessing computer information services. New Jersey: Prentice-Hall, 1984.

Special report: Telecommunications: Bringing the world to your school. Electronic Learning, November/December, 1984.
Shea, G., Information services: The new frontier of communication. 33,34,88,89
Butcher, D. E., On line searching: How to get instant information. 39,40,90
McGinty, T., Buying modems: What to look for. 42,43,92

TELECOMMUNICATIONS

Apple Telecomputing. inCider, August, 1984.
Ryan, B., Don't hang up. 54-57
Granfield, J., Buying your first modem. 58-60
Witham, J., Directory of modem manufacturers. 61-63
Kitzerow, S., Direct connect diary. 64-65

Darling, S., A buyer's guide to modems. COMPUTE!'s Gazette, November, 1984. 30,32,37 (Focus on Commodore)

Sims, R., Gazette Telecommunications Software: Part 1: C/G Term. COMPUTE!'s Gazette, November, 1984. 38,40,44,46,48

BULLETIN BOARDS

Electronic pen pals. inCider, August, 1984. 17

Levitan, A. R., IBM PC Bulletin Boards (updated February 1984). COMPUTE!s PC & PCjr, April, 1984. 74-76

Yakal, K., Bulletin board fever. COMPUTE!'s Gazette, November, 1984. 16,20,24,26,28

LOGO

Teacher Training and Logo: Higher Levels of Concern

C. Dianne Martin
George Washington University
EE & CS Department
Washington, D.C. 20052

Rachelle S. Heller
University of Maryland
Computer Science Department
College Park, MB 20742

Numerous workshops and courses for teacher training in LOGO have been reported upon over the past three years. The success of the courses has varied widely. This paper describes the goals and outcomes of such a course and an analysis of those outcomes using the stages-of-concern taxonomy (Hall, George and Rutherford, 1977). Based upon the implications from this taxonomy, the authors present suggestions for structuring teacher training courses.

INTRODUCTION:

Teacher training has been identified as one of the crucial keys to implementing computers in the classroom (U.S. Office of Technological Assessment, 1982). With the availability of LOGO on microcomputers, school systems are now struggling to adapt their curriculums to accommodate this new computer "culture" (Watt, 1984). LOGO presents the double dilemma of introducing a new skill as well as the possibility of a radically different classroom management style.

Hall and Loucks (1978) have stated that bringing about innovation within education is not an end product but a process. Further, they suggest it is important to focus on the individual who goes through stages of concern when experiencing such an innovation. Bringing LOGO to children in the classroom typifies such an innovation. Figure 1 describes the stages-of-concern taxonomy developed by Hall, George and Rutherford (1977).

In June 1983 the authors were approached by a Washington, D.C. Metropolitan suburban school system to develop and implement a graduate course for educators using LOGO. The course was taught using a guided discovery approach (Martin and Heller, 1984). It was designed with three outcomes in mind - to cover LOGO content, to demonstrate methods of teaching LOGO in a classroom setting, and to impart the underlying educational philosophy of the LOGO culture (Watt, 1983). The course syllabus is shown in Figure 2.

LOGO CONTENT:

The LOGO content consisted of the syntax, semantics, and workspace management of the language LOGO. The content achievement level of each participant in the course was measured by three criteria - weekly programs written in LOGO, a LOGO language exam given on the ninth week, and a final project.

In a guided discovery atmosphere the participants were encouraged to choose programming topics from among a weekly list of suggested activities, or they were free to explore their own ideas. These programs were not given a grade, but the instructors judged and reacted to them based on correctness, modularity and style. The participants were encouraged to select project topics which would allow them to implement LOGO in their own teaching environ-ments.

Based on the three criteria, all of the participants achieved the ability to write and understand linear LOGO turtle graphics programs. Half of the participants were able to handle programs with variables, single-level subprogram calls, and iteration. Very few were able to use and understand multi-level subprogram calls and recursion.

The outcome of LOGO content proficiency addressed only the first three levels of the concerns based taxonomy; that is, the self-directed levels of the taxonomy. All of the participants reached at least the information level of concern. Those particpants who became concerned with mastering LOGO concepts for their own intellectual satifaction could be said to have moved to the personal level of concern.

TEACHING METHODS:

Proficiency in methods of teaching proficiency was judged by comments collected on weekly evaluation forms, by class discussion, and by the thrust of the final projects. Since the course was taught with a guided discovery approach, it was expected that participants would model and focus on LOGO teaching methods and classroom management techniques. Using the master teacher model, classroom discussion, project planning time, and the laboratory environment, it was hoped that participants would learn LOGO teaching methods through their own learning experience. The fact that this proved to be an unrealistic expectation for the majority of participants can be explained by the stages-of-concern taxonomy.

The innovation in this course was the use of LOGO with children in the classroom; however many of the participants did not address the innovation at the task-directed or management level of concern. In spite of the opportunities provided to deal with this concern, less than half of the participants were ready to actively deal with it during the time frame of the course.

EDUCATIONAL PHILOSOPHY:

This course was structured to provide weekly class discussions in hopes that participants could be encouraged to reflect upon the LOGO educational philosophy to its full ramifications. The outcome concerning educational philosophy can be related to the last three levels of the stages-of-concern taxonomy. Out of the two semesters only three teachers showed concern at the consequence level. Only two participants, the original protagonists for the course, showed concern at the collaboration level. No participant was observed to have exhibited the refocusing level of concern.

CONCLUSIONS:

Further changes need to be incorporated which affect the methods and philosophical outcomes of the course. Future sessions of the course should include the opportunities and encouragement for all participants to begin to use LOGO with children during the course (Cohen and Elkin, 1984). To address the outcome of producing change in educational philosophy, teachers need to have the opportunity to become part of and have a vested interest in a LOGO sharing network. Further, teachers need to be trained to observe and reflect upon their teaching style and classroom environments. Subsequent offerings of the course should include a component to teach observation techniques and provide observation opportunities for the participants (Kull, 1984) to focus on intervention and collaboration styles of teachers and students.

Experience from this and other teacher training courses (Philip, Muntner, Cutlip, 1982) has shown that it is unrealistic to expect to move all participants through all levels of the stages of concern taxonomy. Teacher training should therefore be designed to create an atmosphere which will allow the participants to attain

LEVEL	CONCERN	DESCRIPTION	FOCUS
0	Awareness	Coming into contact with the innovation	Self-Directed
1	Information	Wanting to learn about the innovation; fact gathering	
2	Personal	Concern with the impact of the innovation on self	
3	Management	Using the innovation in context of immediate environment	Task-Directed
4	Consequence	Concern with effect of the innovation on immediate environment	Impact-Directed
5	Collaboration	Interacting with others involved with the innovation	
6	Refocusing	Refining and disseminating the use of the innovation with others	

Figure 1: Stages-of-Concern Taxonomy

WEEK	TOPIC
1	Using a Computer to Solve Problems
2	How Computers are Used in Education
3	LOGO as a Procedural Language
4	LOGO Graphics
5	Use of Variables in LOGO
6	String Manipulation Capabilities in LOGO
7	Random Selection and Recursion in LOGO
8	LOGO Geometry
9	LOGO Language Exam
10	Looking at LOGO resources
11	Project Development Lab
12	Project Development Lab
13	Social Impact Lecture; Presentation of Projects
14	Round Table Discussion - LOGO Implementation

Figure 2: Syllabus for Introduction to LOGO for Educators

whatever level of concern is appropriate for them. Daniel Mellman, computer education expert within the Kibbutz movement in Israel, compares the untrained teacher to one sitting in the dark unable to move. There are varying degrees of light, but even with a little light to dispel the darkness, one can make progress. The goal of concerns -based teacher training should be to provide each teacher with at least a candle.

REFERENCES:

(1). Cohen J.. and Elkin R. "LOGO for Primary Grade Teachers-An Inservice Model." (Ben Gurion University of the Negev, Project for Computing in Education & Israel, July, 1984).

(2). Hall, G.E., "Issues Related to the Implementation of Computers in Class-rooms: Where to Now?", (Research and Development Center for Teacher Education, University of Texas at Austin, February, 1981).

(3). Hall, G.E., George A.A. and Rutherford, W.L., "Measuring Stages-of- Concern about the Innovation," (Research and Development Center for Teacher Education, University of Texas at Austin, 1977).

(4). Hall, G.E. and Loucks, S., "Teacher Concerns as a Basis for Facilitating and Personalizing Staff Development," (Teachers College Record, Vol. 80, No. 1, September, 1978).

(5). "Informational Technology and It's Impact on American Education (Summary Report)," (Office of Technological Assessment, Congressional Board of the 97th Congress, Washington, D.C., November, 1982).

(6). Kull, J., Learning and LOGO in G.Fein and P. Campbell (eds) Microcomputers in Early Childhood Education: Conceptualizing the Issues, (Reston, 1984).

(7). Martin, C.D. and Heller, R.S., "Status Report: Inservice Computer Literacy Training," Informatics in Elementary Education, (IFIP Working Conference, North Holland, Amsterdam, 1983).

(8). Martin, C.D. and Heller, R.S. "Teaching LOGO to Teachers: A Look at the Issues," Informatics and Teaching Training, (IFIP Working Conference, North Holland, Amsterdam, 1984).

(9). McCarthy, B., "Improving Staff Development through CBAM and 4 Mat Tm ", (Educational Leadership, October, 1982).

(10). Papert, S., Mindstorms, (Basic Books, New York, 1980).

(11). Philip, C.C., Muntner, J. Cutlip, P. "Computer Literacy Education for K-6 Teachers." (Proceedings of the 1982 AEDS Convention, Orlando, Florida, May, 1982).

(12). Watt, D., Learning with LOGO, (McGraw-Hill, New York, 1983).

Creating Logo Microworlds

Daniel H. Watt
Educational Alternatives
Gregg Lake Rd.
Autrim, NH 03440

This talk demonstrates the use of Logo as a programming environment for creating *microworlds*, small, exploratory, computer-based learning environments. The microworld approach confronts two key issues for Logo-using educators: how to incorporate Logo activities into the curriculum, and how to make use of Logo's potential as an intermediate and advanced programming language.

My goal in this talk is to demonstrate the usefulness of Logo as a programming environment for creating *microworlds*, small, exploratory, computer-based learning environments that can be used in a broad range of educational settings.

Logo was created to be both a computing language and a learning environment. Although it was designed to have "no threshold and no ceiling" (Papert, 1980), that is, to be suitable for advanced programmers as well as beginners, Logo's use in educational settings has been limited mainly to children in the elementary and middle school grades. In addition, most Logo use has been limited to turtle graphics, ignoring Logo's potential for interactive programming and information processing.

Last summer I taught two graduate level courses for teachers which had as their major focus, the design and development of microworlds. In the talk I will be presenting examples of Logo microworlds created by these teachers, as well as ones I have developed myself.

The idea of a *microworld* is not a well-defined concept, although the term is beginning to be widely used. Seymour Papert has described microworlds as "incubators of knowledge" -- environments in which children can acquire powerful ideas by carrying out activities within them. The best known example is the Logo microworld of the turtle, in which important mathematical ideas can be learned through exploration (Papert, 1980).

Robert Lawler, one of the first researchers to develop microworlds along the lines advocated by Papert, has offered a set of criteria for microworld design. In his view a microworld should be based on "'neat phenomena' -- phenomena that are inherently interesting to observe and interact with." Furthermore, it must include "powerful ideas." According to Lawler such ideas are *simple* enough that they can be easily understood by the learner; they are *general*, having application

BEST COPY AVAILABLE

outside of the microworld itself -- otherwise why bother creating it; they are immediately _useful_ to the learner -- otherwise why bother learning them: and they are _syntonic_, closely related to the learner's existing mental models (Lawler, 1982b).

Microworlds, whether on or off a computer, can take many forms. They can be games, simulations, programming languages, tools for creative expression, or environments (like sandboxes) that encourage playful exploration. One distinction that is important for me, however, is what microworlds are _not_: microworlds are _not_ computer-based lessons, tutorials, or games designed to lead learners to particular sets of facts, skills or conclusions in a structured sequential manner.

In addition, microworlds are not universal in their application. A phenomenon that is "neat" for me may turn out to be "boring" for you. An idea that you find to be powerful and important, may be so far beyond my grasp or so unrelated to my interests and concerns, that it is neither simple nor useful for me. One of the most successful microworlds has been the world of turtle geometry. But most people who teach Logo have found some learners for whom the turtle's world is simply uninteresting. So the microworld designer must take into account the interests and knowledge of the particular learner or learners he is designing for. Lawler, for example, has designed most of his microworlds primarily for his own children, drawing on their particular interests, environmental setting, and learning needs for inspiration (Lawler, 1982a, 1982b).

Logo-based Microworlds

I'd like to describe some examples of my own microworlds, and those of teachers who have worked with me and with Molly Watt. I hope that readers will be inspired to create examples of their own. The examples include several different kinds of applications, and have been constructed using different levels of Logo programming skills.

Word-action microworlds are among my favorites. The idea was first articulated by Seymour Papert and developed by Lawler (Lawler, 1982a,b). They are designed for use by very young children, having their first encounters with written words.

The idea is to create an environment that parallels the way children learn to speak by issuing commands. The word "Mama" or "Daddy" brings a parent from the other side of the room. "Up" means "pick me up and carry me." Saying the words "milk" or "juice" will often result in a drink. And so on.

In a word-action microworld typing a word on the keyboard produces an action on the TV screen. Current examples are based on the use of sprites -- small colorful objects that can take on a variety of shapes and move freely across a computer screen. One example, the Farm microworld, was created in the summer of 1984 in Vancouver by David Bell, Leon Lebrun, Larry Wiebe and Elaine Willis for Bell's four-year-old son, who

BEST COPY AVAILABLE

lived in a farming community and was fascinated by farm animals and equipment. If the child types a name such as PIG, COW, TRACTOR or SUN, an object with the appropriate image appears on the screen. Action words such as WALK, DRIVE, FLY, UP, DOWN, LEFT or RIGHT, set an object in motion or change its motion. WHITE, YELLOW, RED, GREEN, or BLUE change its color. "Reality" is not a limitation. In this world pigs can fly and trees can walk -- but only if the child wants them to.

The objects and vocabulary of a word-action world must be tailored to the individual user. Gerri Sinclair, also in Vancouver, built a microworld full of city vehicles, cars, trucks, planes and trains, for her two-year-old son. Bob Lawler used a BEACH setting for his children, who lived near the seashore. The important thing is that the same Logo procedures can be used for any number of worlds, just by changing the names and the shapes. So although the size and complexity of current examples are limited by memory limitations, it's easy to make lots of different ones, that can be used one at a time.

The Logo Poet

The Logo poet is one of many microworlds oriented around generating language patterns using parts of speech. The version I will show generates "poems" that sound something like Haiku (Watt, 1983). Other possibilities involve random sentences, conversational programs (Abelson, 1982), or language games such as Madlibs (Watt and Watt, 1985). Here is a sample (Watt, 1983):

EACH LIMPID POND
ONE BIRD RACES OVER
  THE FOSTY FIR
WILD BLUE MOON

Each example follows the same pattern of nouns, verbs, adjectives, and so on, but with different words, chosen randomly from long lists of words. To change the results, alter the pattern or substitute your own word list. In this environment, you can explore the structure of language, the cultural values of different people, and the usefulness of parts of speech when used to construct human expression, rather than simply to analyse it.

Programs like this can be used for fairly sophisticated language explorations. A college class has used it for a close analysis of traditional Haiku forms (Hoffer and Sammes, 1984). Others have made the selection process more complex so that the words chosen for a given poem are related to each other in predetermined ways (Sharples, 1980). The possibilities have only begun to be tapped.

Learning Logo Itself

A number of microworlds have been developed to help beginners learn Logo itself. Rather than have to deal with all of Logo's complexity, such environments simplify Logo, making it a less of a challenge, and providing more focus. But when properly designed, such microworlds do not replace the full Logo environment, but lead a learner into it more gently, or help a learner master some complicated aspect of turtle graphics.

Examples will be demonstrated that allow young children to control the turtle using single keys (Watt, 1983, Abelson, 1982), games to help learners build confidence in estimating angles and distances, and activities that help children learn their way around the turtle's screen by placing preprogrammed shapes wherever they want them. Other microworlds let people explore the mathematics of turtle geometry, highlighting such areas as polygons and stars, symmetry, rotation, pattern making, tessellations, and recursive fractal designs.

The last set of microworlds I will demonstrate are a set of games in which the turtle has been programmed to behave in accordance with Newton's laws of motion. These games allow students to develop their intuitive notions of the movement of objects in space, without having to master a set of complicated equations and formulae.

Microworlds and the Future

I believe that microworlds have an important place in the future of education. They provide educators with ways to informally extend the richness of the classroom environment while incorporating computers into the ongoing curriculum. Although the process of microworld development has barely begun, I can foresee a time -- perhaps 3 to 5 years from now -- when there will be enough educators with knowledge of Logo, and a culture rich enough to support sharing of ideas and programs, so that microworlds can become an important part of the learning process.

References

Abelson, Harold, *Logo for the Apple II*, Byte Books / McGraw-Hill, 1982.

Hoffer, Bates L. and Pat Semmes, "Haiku and Nim: Logo in the Language Arts," *Proceedings, National Educational Computing Conference*, Dayton Ohio, 1984.

Lawler, Robert W. a) "In the Lap of the Machine," *Boston Review*, June 1982.

b) "Designing Computer-based Microworlds," *Byte*, August, 1982.

Papert, Seymour, *Mindstorms*, Basic Books, 1980.

Sharples, Mike, "A Computer-based Language Workshop," *Sigcue Bulletin*, Vol. 14(3), 1980.

Watt, Daniel *Learning With Logo*, Byte Books/McGraw-Hill, 1983.

Watt, Molly and Daniel Watt, *Teaching With Logo*, Addison-Wesley, 1985.

BEST COPY AVAILABLE

It's Almost Time For Logo

J. Dale Burnett
Queen's University
Kingston, Ontario

Many people within the educational computing community are championing Logo to play a major role in the future. While generally sympathetic with this thrust, I believe that one must also recognize the influence of substantive limiting factors. Nonetheless, we appear to be "facing the right direction".

In 1973 McLean prepared a publication entitled, "It's Almost Time for CAI". The article was unabashedly pro-computer assisted instruction, but it also recognized that difficulties with hardware capability and software compatability, combined with an inadequate instructional theory, meant that widespread implementation was still in the future.

Twelve years have passed. The advent of the microcomputer has gone a long way toward solving the hardware capability problem. Many of today's machines are technically capable of satisfying an instructional designer's desires. Software compatability does not appear to be as large a problem as foreseen just a few years ago. Most commercial software developers are able to provide equivalent packages for many different machines. Other developments are occuring that permit one machine to "play" programs intended for a competitor's machine. The third impediment, inadequate instructional theory, continues to be a cause for concern. This is a limiting factor but not a critical one. Thus we are able to pursue our instructional goals, perhaps in some sub-optimal fashion, without having to wait for strong theoretical guidance. In fact, such developmental efforts may well contribute to such a theory.

Ten years ago there were many individuals who were strongly committed to the widespread development and incorporation of CAI. Today a new sub-group is identifiable: the Logo missionaries. In many respects I would like to be counted among them. Stereotypic misssionaries have certain admirable traits. High among them is a sincere, well-intentioned commitment. On the other hand there are weaknesses when this zeal is carried to excess - most notably expressed as an intolerance for other views or approaches. For many the time for Logo is now. For me, it's almost time for Logo.

First, it may be helpful to note a few factors that have helped contribute to the current wave of Logo enthusiasm. In order for any computer-based approach to be a serious contender for school use, it must be readily available for machines that are now, or are soon to be, in school environments and at a cost that is affordable. There are versions of Logo now available for most microcomputers now being marketed by Apple, Atari, Commodore, IBM and Radio Shack. Prices vary but almost all are under $200. Existence is necessary but not sufficient for widespread adoption. One must also want to use it. This is closely related to ease-of use and to power. Logo was designed with young children in mind. The syntax permits one to begin using Logo within a few minutes of its first introduction. However it is a mistake to equate this feature with the phrase "a kiddie language". The language is sufficiently powerful to engage serious study at a graduate mathematics level - witness, for example, the excellent book by Hal Abelson and Andrea diSessa, Turtle Geometry . Logo enthusiasts refer to these two features as "low threshold, high ceiling". It is worthwhile noting that natural language also has a low threshold and a high ceiling.

Another important feature for any new software product is that it be "close to the concept". Given that one is interested in a particular topic, the software should facilitate thinking about the topic. One should not have to think like a computer. This point has been forcefully made on many occasions by Ken Iverson, the original developer of another computer language (APL) that merits serious consideration by mathematics educators. Logo also provides a close correspondence between the syntax of the language and the ideas one is trying to explore, particularly when one is constructing geometric patterns.

A final consideration on the positive side of the ledger is marketing. This is a subtle factor involving not only direct advertising but the diffusion of ideas through a variety of means. The publication of Seymour Papert's book Mindstorms occurred at the same time that microcomputers first became widely available and at the same time that the first versions of micro-based Logo were appearing. Timing is critical, a theme to which I will return. Diffusion of ideas also is strongly influenced by research efforts. Educational researchers around the world have recognized Logo as a potentially fruitful area of research, particularly by individuals with a strong interest in cognitive science. Thus the topic and the issues remain alive. The combined effect of the above factors is substantual.

Before continuing the discussion of additional reasons why I believe Logo is a prime candidate for increased use in our schools, I would like to identify a few reasons why, although optimistic, I believe we are still a few years away (perhaps ten !!) from witnessing its widespread use. Five limiting factors will be identified: (1) Public education has, and I believe should, remain teacher-intensive. Thus any substantive change to the system must take into consideration the enormous number of people involved (Burnett, 1984). (2) Curriculum guidelines and resources need to be revised and developed on a much faster time-line than has been necessary in the past. (3) The profession needs to develop more effective mechanisms for both administrative and peer support. (4) Societal and political pressures may dictate increased emphasis on emerging technologies but they are still likely to leave much of the detail to the educators. (5) It is not clear what role programming in general is likely to play in the future. In large measure the issue is not Logo, nor even computers, but the more general notion of the wide-spread implementation of innovation and change.

Simple cause and effect relationships are rare when one considers societal trends. Nonetheless there are often occassions when a variety of relatively independent factors all seem "to be pointing in the same direction". This is the issue of timing referred to earlier. Three developments have already been mentioned. The development of the microcomputer brought the capabilities of the computer into the classroom and into the hands of individual students, Logo was developed specifically for children and Mindstorms was published - which stimulated discussion on underlying educational philosophies.

Two additional books, one on education and one on business appeared at about the same time with similar messages. The educational book was written by a former Dean of The Graduate School of Education at Harvard University, Theodore Sizer. This book, Horace's Compromise , is the first report emanating from a nation-wide study of high schools in the United States. The following quotations were selected to give an overview that is strongly similar to much of the current Logo literature: "Teachers without judgement stumble. It is the heart of teaching ... I want hungry students and hungry teachers ... Above all , the hungry student is active, engaged in his or her own learning ... The student takes the initiative and works at teaching himself. Lamentably, far too few American adolescents are hungry students ... There are too few rewards for being inquisitive ... We assume they cannot figure things out for themselves, so we tell them things ... 'Ownership' of an idea assists in its retention ... We all like our own ideas ... There is little joy in accurate regurgitation ... Human factors rather than physical ones most shape the climate of a school".

The second book was Peters and Waterman's best seller, In Search of Excellence . Once again a sampling of quotations is revealing: "extraordinary energy exerted above and beyond the call of duty when the worker is given even a modicum of apparent control over his or her destiny ... scale efficiencies give way to small units with turned-on people ... attention to employees, not work conditions per se, that has the dominant impact on productivity ... They encourage practical risk taking, and support good tries ... Make sure you generate a reasonable number of mistakes ... interact, test, try, stay in touch, learn, shift direction, adapt, modify, and see ... to communicate a sense of excitement, fun, and zest in all that we were doing".

The challenge facing the educational community is to review our perspectives and our approaches in order to improve the preparation of our graduates for a rewarding future. To the extent that many of the above quotations are "close to the mark" - and they are generally consistent with the findings of a recently completed two year study funded by the Ontario Ministry of Education on the effects of the creative uses of microcomputers (with a particular emphasis on Logo) in elementary classrooms (Carmichael et al., forthcoming) - Logo oriented activities appear to be a step in the right direction. Clearly more analysis is needed, and other approaches tried as well, but we must be careful not to imitate the centipede about to take its first step.

References :

Abelson, H. & diSessa, A. (1981). *Turtle Geometry* . Cambridge, Mass.: MIT Press.

Burnett, J. D. (1984). Logo for teacher education. In M. Yazdani (Ed.) *New Horizons in Educational Computing* . Chichester: Ellis Horwood.

Carmichael, H. W., Burnett, J. D., Higginson, W. C., Moore, B. G. & Pollard, P. J. (1985). Computers, children and classrooms: A multisite evaluation of the creative use of microcomputers by elementary school children. Forthcoming.

Iverson, K. E. (1980, August). Notation as a tool of thought. ACM Turing Award Lecture, Communications of the ACM, 444-465.

McLean, L. D. (1973). It's almost time for CAI. An informal publication of the Department of Computer Applications, The Ontario Institute for Studies in Education, 24pp.

Papert, S. (1980). *Mindstorms* . New York: Basic Books.

Peters, T. J. & Waterman, R. H., Jr. (1982). *In Search of Excellence* . New York: Harper & Row.

Sizer, T. R. (1984). *Horace's Compromise* . Boston: Houghton Mifflin.

Teaching Recursion In Logo By Playing Turtle And Playing Procedure

Molly Watt
Educational Alternatives
Gregg Lake Road
Autrim, NH 03440

Teaching the powerful concept of recursion is a fundamental job for teachers using Logo, as recursion is an essential part of the way Logo programming works to solve problems. Since recursion is an abstract mathematical concept, it often becomes a stumbling block for teachers and learners alike. This paper offers procedure plays as a tool for providing a concrete image of recursion, lightening this aspect of learning to use Logo, and as a tool which can make any Logo procedure conceptually clear.

"Recursive is a mathematical term for a repeated procedure..."
(Poundstone, 1985)

Recursion is one of the most powerful ideas embedded in the Logo language. Being able to use recursion is fundamental to being able to use Logo as a programming language. It is also one of the most difficult concepts for most people to learn. In the same way that "talking to the turtle" provides a metaphor for making the abstract activity of turtle geometry concrete, playing turtle and creating procedure plays provides the learner with concrete models for primarily abstract processes.

As a young girl I was intrigued with a picture in a magazine. It showed a girl about my age holding a picture of herself holding a picture of herself holding a picture of herself... The picture never finished, it simply became too small for my eyes to differentiate.

The idea of recursion is simple: There is a big job to do such as all the dishes for a birthday celebration. The usual way the dishes get done is by the parent of the birthday person after all the guests have departed! But if Logo could leave the computer and take on the job, it would use a recursive procedure to complete the task in a much less burdensome manner. Study the following example:

```
TO DO.BIRTHDAY.PARTY.DISHES
  If your placemat is empty
    STOP
  Otherwise: clear dishes from
    placemat
  Wash dishes
  Dry dishes
  Put dishes away
DO.BIRTHDAY.PARTY.DISHES
END
```

We can consider each Logo procedure a script for a play in Logo (Watt and Watt, 1985). Using the above procedure as a play script, the first player takes the role of DO.BIRTHDAY.PARTY.DISHES. The first line of the script directs the player to check to see if the placemat is empty. If there are no dishes, the play is over at the end of the scipt line. If, however, there are dishes on the placemat --

BEST COPY AVAILABLE

and in our example, there are -- the player continues to follow the procedure script by following its instructions line by line. First the player clears the dishes, then washes the dishes, then dries the dishes, then puts them away and then calls the next player to follow another copy of the procedure script. The next player acts out the same instructions and calls another player who follows a new copy of the same instructions. In this way the birthday party dishes are completed by each player in turn following a procedure which includes calling the next player until eventually a player is called who has an empty placemat and the procedure stops and the play ends.

In the same way students can act out a procedure play of DO.BIRTHDAY.PARTY.DISHES as an example of Logo's wizardry at work. They can begin to understand recursion by playing turtle in a simple tail recursion Logo procedure.

How do you teach the turtle to make a circle in Logo? Well, the simplest answer in English is to tell the turtle to "move forward a little and turn a little." This command can be given to a student playing turtle and walking out the instruction on the floor. The instruction can be given again and again until eventually the student has walked a circle.

Translating the same example into a logo procedure would look like this:

```
TO CIRCLE
  FORWARD 1
  RIGHT 1
CIRCLE
END
```

The image is that the first copy of the Logo procedure CIRCLE moves the turtle FORWARD 1 and then RIGHT 1 and then calls for a new copy of the CIRCLE instructions which move the turtle FORWARD 1 and RIGHT 1 and then calls for a new copy of the CIRCLE instructions until the process is interrupted. Note that the instruction "END" is never reached in CIRCLE as a new copy of CIRCLE is called just before END.

A more complex script for a procedure play is:

```
TO COUNTDOWN :NUMBER
  IF :NUMBER = 0 STOP
  PRINT :NUMBER
COUNTDOWN :NUMBER-1
END
```

(Abelson, 1982)

In this play the first player, COUNTDOWN receives a number input of any number. For the sake of a simple written example I'll use a number 2. (In a workshop of people acting out procedures a larger input number is more fun to use!)

COUNTDOWN carrying 2 checks to see, is 2 = 0 ? It's not. COUNTDOWN carrying 2 goes on to the next line and follows the instruction to print its number, a 2, on the space provided. (I use a pad of paper to simulate the computer monitor.) Then COUNTDOWN carrying 2 subtracts 1 from its 2 and calls a new player COUNTDOWN and hands over an imput of 1.

COUNTDOWN carrying 1 checks to see, is 1 = 0 ? It's not. COUNTDOWN carrying 1 goes on to the next line and follows the instruction to print its number, 1, on the space provided. Then COUNTDOWN

BEST COPY AVAILABL

carrying 1 subtracts 1 from its 1 and calls a new player COUNTDOWN and hands over an input of 0.

COUNTDOWN carrying 0 checks to see, is 0 = 0 ? It is. So COUNTDOWN carrying 0 stops and calls "I'm done" to COUNTDOWN carrying 1 who calls "I'm done" to COUNTDOWN carrying 2 who tells the audience "I'm done."

The play is over and applause is in order. After the applause has died down, execute the same procedure on the computer.

A more complex procedure illustrating embedded recursion is MYSTERY.

```
TO MYSTERY :NUMBER
  IF MYSTERY = 0 STOP
  MYSTERY :NUMBER-1
  PRINT :NUMBER
END
```

The mystery of this classic Logo procedure is answering the question about what the procedure does (Abelson, 1982). Ask students to predict what they think will happen before they perform the play. Then by actually performing this procedure as a play, an image for how embedded recursion works will be clear, logical and remembered by all play participants and some of the audience as well! All the players need to do is follow the instructions and understanding will emerge in the process of performing.

References

Abelson, Harold, Logo for the Apple II, BYTE Books/McGraw-Hill, 1982

Anderson, Peter, Logo Processes :ALIST, IN LOGO PARENTIS PUBLICATION, 1984

Hofstadter, Douglas, Godel, Escher, Bach, Basic Books, 1979

Poundstone, William, The Recursive Universe, William Morrow, 1985

Riordan, Tim, "Helping Students with Recursion: Teaching Strategies I, II, III," The Computing Teacher, December/January, 1983-4, February, 1984, March, 1984

Stavely, Tony, "List Processing in Logo," The Computing Teacher, December, 1982

Watt, Daniel, Learning with Logo, McGraw-Hill, 1983

Watt, Molly and Watt, Daniel, Teaching with Logo, Addison-Wesley, 1985

BEST COPY AVAILABLE

The OISE Logo Project: Apple/Terrapin Software

Rina Cohen and Graham Barker
The Ontario Institute for Studies in Education
252 Bloor Street West, Toronto, CANADA M5S 1V6

A set of Logo-based microworlds, games and utility packages will be presented and discussed.

This session will demonstrate and discuss the development and use of a set of Logo-based microworlds, games and utility packages, for the Apple computer. These packages were developed at OISE for use as Logo resource materials in elementary classrooms, providing links between existing curricula and teaching/learning techniques and those that Logo offers. Although the project has formally ended, the materials are still being field-tested in a number of schools and are not available for general distribution.

In 1982-83 we were invited to observe teachers from the North York Board of Education, in Metropolitan Toronto, who were using Logo in their grade 2 and 3 classrooms. Our observations supported the view that in teaching with Logo, teachers would benefit from a variety of Logo programs, teaching materials, class activities and other resources, to allow children a range of appropriate choices of activity. Because Logo has only recently begun to be introduced into schools, this rich variety of support materials is not yet available. Through this project, carried out during 1983-84 in cooperation with seven teachers in the North York Board, we have taken our first step in contributing to such a collection of resources.

The materials provide a varied and rich set of learning possibilities in an attempt to integrate the demands of the structured curriculum with the open-endedness of the full Logo language.

Learning Purposes and Types of Packages Developed

The packages have been designed with three different types of learning environment in view, although often these categories overlap with fuzzy boundaries:

1. *Microworlds*: Process-oriented "problem-spaces" or "playgrounds" in which some interesting things can happen. Within the boundaries of these worlds there is freedom for in-depth and open-ended exploration. Examples include: Sticks&Blocks, Singing Sticks&Blocks, Sticksongs, Chatter, Facelist.

2. *Games*: Goal-oriented, well-defined objectives, playing instructions, scoring rules and a more limited range of options for learning, practice or exploration. The game packages include: Turtle Trails and Turngames.

3. *Utilities*: Product-oriented tools which are more general in scope than microworlds or games and can be used for a variety of purposes, e.g.: Moviemaker, Drawing Kit, Chatter Editor, Stick.Songs.

Several kinds of learning purposes are encouraged through the use of these packages. They may be isolated or integrated and no instructional sequence is implied by the order in which they are listed here:

1. The development of sub-skills or basic concepts concerned with: *spatial orientation* (right-left discrimination, reversal); *mathematics* (angles and angle rotation, measurement, additivity and area); *language arts* (structure

and meaning of English vocabulary and sentences, oral and written description and expression of ideas); *music* (basic notions of pitch and duration, making simple melodies); *visual arts* (design and arrangement of coloured graphic patterns or pictures); *problem-solving* (estimation, guessing, planning, modelling, testing and debugging); and *basic Logo programming skills* (use of Forward, Backward, Right, Left and Repeat commands, and the list processing operations).

2. Basic Logo skills including knowledge of some of the built-in "primitive" commands in Turtle Graphics and in list processing,

3. Key programming concepts including describing, building, naming, using and fixing small programs (called"procedures" in Logo).

4. Attitudes, skills, and concepts involved in problem-solving, including guessing, setting goals, planning actions, testing them out, debugging and revising plans or actions.

5. Play and creative expression with language, graphics and sounds.

6. Learning how to make effective use of a given set of tools (called "utilities").

7. Enhancing individual self-concept as well as social and communication skills through the activities in the packages.

Brief Descriptions of the Packages

The summaries given below provide an overview of the packages.

STICKS&BLOCKS
A microworld intended for beginning Logo users, which allows exploration of the basic concepts of distance, forward and backward movement, height and width of two dimensional boxes, left and right orientation and of repetition. (No commands are included for angle or angle rotation). These concepts are explored via the production of colored sticks and color-filled boxes in a variety of patterns, with a simplified set of turtle graphics commands and and interactive BOX and REPEAT commands. Grade levels K-4.

SINGING STICKS&BLOCKS
This package adds the dimension of sound and music to Sticks&Blocks. The concepts of pitch and duration can be explored by making sticks and boxes which include simple sound effects. The height dimension of the sticks or boxes is associated with the pitch of the tone, while the width dimension is in direct relation to the duration of the tone. Musical designs can be made by arranging sticks and boxes, based on a chromatic scale in which each vertical unit is a semi-tone. Grade levels K-4.

STICKSONGS
A simple music composition system which emphasizes exploration of some basic musical notions through the creation of simple graphics and musical compositions. The user can develop an understanding of pitch (and in a limited sense, duration) by experimenting with sticks of different heights, each height representing a different pitch. Stick.Songs differs from Singing.Sticks in that it is specifically designed to allow easy and immediate creation and exploration of simple melodies in a chromatic scale with its music editing features. Grade levels 1-6.

DRAWING KIT
This package provides the user with a simple way to create drawings on the screen, using a simplified version of the basic turtle graphics

commands, plus some ready-made objects (circles, squares, triangles, houses, birds, sun) which can be placed on the screen at locations specified by the user. Some of these can be filled with colour. The user can also modify the procedures used in the package. Grade levels K-2, Special Education.

MOVIEMAKER}
Moviemaker is a utility package which integrates text and graphics. Users can design or read in pictures, add text to each picture, then save a series of pictures with their text in sequence. The pictures can then be run as a "Movie" (or a "slide show") or printed as a book, using single-word commands. The package is menu-driven and includes instructions. Grade levels 3-8.

TURTLE TRAILS (3 packages)
In these games the player has to direct the turtle over increasingly difficult pathways or trails. Three difficulty levels are included in three separate packages. They are designed to assist beginning Logo users and younger children in learning about direction and distance as related to the angles of rotation in the turns and the length of the trails. The packages provide instant feedback, graphics rewards and running scores. Grade level K and up.

TURN GAMES (3 packages)
The Turn Games provide a restricted environment for focussing on one activity: estimating the amount of angle rotation required to turn the turtle towards a given target. Three difficulty levels are included in three separate packages. The object of the game is to turn the turtle until it points to a spot shown on the screen outside the circle. The spot varies randomly with each successive round of the game. The circle is marked by hatches at increments of either 30 or 10 degrees, helping the player aim the turtle. A running score is kept as the player turns the turtle towards the spot. Grade levels K-5.

FACELIST and FACELIST TUTOR
The Facelist microworld allows users to practice the LOGO list processing commands FIRST, BUTFIRST, LAST, BUTLAST. With these operations users specify features (eyes, ears, nose, mouth, hat) for any or all of five faces which form a pictorial list. The Facelist tutor introduces the above list processing commands using a tutorial demonstration approach. Grade levels 4-adult.

CHATTER and CHATTER EDITOR
The Chatter package is intended to encourage play with, and thinking about, language structure and meaning, rather than to provide drill and practice on grammar or vocabulary alone. Chatter generates random sentences according to a given sentence structure pattern and list of words organized by part of speech. The user can "teach" the program new words at anytime.

The Chatter.Editor package provides editing features for creating, modifying and saving the small data bases of lists of words and sentence patterns to be used in Chatter. The user can store these small data bases on personal disks. Teachers can use these packages to complement content teaching in other classroom activities than "language arts". Students can use them to explore on their own. Grade levels 6-8.

Developmental Studies of Children Learning Logo Programming: The Cognitive Dimension

Rina Cohen
The Ontario Institute for Studies in Education
252 Bloor Street West, Toronto, Canada, M5S 1V6

While most of us believe in Logo's educational potential, not much is known yet about how to realize this potential. Collaborative long term research efforts involving teachers, researchers and psychologists will help us determine effective implementation strategies for Logo environments so as to maximize the associated cognitive and educational benefits.

Introduction

Many claims have been made regarding the educational benefits associated with Logo programming (e.g., Papert, 1980; *Byte*, August 1982; *Classroom Computer News*, April 1983; *Computing Teacher*, November, 1982 & December-January, 1984). On the cognitive level, Logo programming activity is said to enhance metacognitive skills such as awareness of one's own thinking processes, planning, strategic problem solving, general mathematical thinking as well as a variety of specific mathematical concepts, spatial orientation, communication skills, and others. In the affective domain, benefits believed to be associated with Logo include, among others, enhanced self concept associated with a sense of mastery and accomplishment, educational autonomy and improved attitude towards learning. Evidenty, the cognitive and the affective effects of a Logo environment are interrelated and to a certain extent, inseparable. For instance, high motivation and a sense of control often lead to better learning and hence mastery of concepts or skills. On the other hand, if for some reason the child is unable to gain mastery over the concepts learned (e.g., because they are beyond her ability level, or introduced pre-maturely), she is likely to get discouraged and loose her sense of mastery and motivation. For this reason, it is important to examine Logo's effects in both the cognitive and affective domains. This paper will deal with both, but with an emphasis on the cognitive aspects.

Research Directed at Evaluating Logo's Cognitive Effects

In view of the great educational potential claimed for Logo, and its rapidly spreading use within the educational system, educators, researchers, adminstrators and decision makers have become aware of an urgent need for concrete evidence supporting the above claims. Only "hard data" substantiating important learning outcomes resulting from exposure to Logo would justify large budgets for Logo implementation.

A number of research studies have attempted to validate the above claims. However, so far, evidence supporting these claims is mostly based on anecdotal information obtained from observational studies documenting individual children's programming activities and some of the motivational and social effects of the Logo environment (e.g., the Brookline project, Papert, Watt, et al., 1979; Statz, 1973; Solomon, 1982). Not much has been achieved as yet in terms validating of the important claims made about Logo's cognitive effects. This might be accounted for in part by the fact that general skills such as problem solving and planning and their transfer to other domains are notoriously difficult to assess (Krasnor, & Mitterer, 1983). A few studies focussing on the development of specific mathematical or spatial skills have been somewhat successful (e.g., Milner, 1973, Shultz, et al., 1984). Particularly, in a series of studies at Edinborough (Howe, 1980, 1981), levels of mechanical aptitude, mathematical problem solving ability and reasoning skills have been identified as key determinants of progress in learning Logo programming. However, the teaching strategy used in these studies was highly structured and curriculum-based, as opposed to the free discovery approach advocated by Papert.

Recently, Pea and Kurland at the Bank Street College of Education (Pea, 1983; Pea, & Kurland, 1983; Pea, 1984) have undertaken a series of investigations in order to provide systematic data on children learning to program in Logo and the alleged cognitive outcomes of such learning. The results of their studies did not support any of the above claims for cognitive benefits; in particular, the children exposed to Logo did not display greater planning skills than the control group, and observations suggested that "problem solving skills may need to be taught directly rather than to emerge spontaneously from learning Logo." (Pea, 1983, p. 26). Moreover,it was found that 8-12 year old children, after a whole year of exposure to Logo programming, following the discovery learning pedagogy advocated for Logo, have not mastered such fundamental programming concepts as iteration, branching and recursion. Their understanding of these concepts was highly context-specific, rote in character and did not transfer to other relevant programming tasks.

Similar results were reported by other recent studies. In particular, Hillel and Erlwanger (1983) and Leron (1983) observed that in the absence of structure and substantial adult support, only a rather shallow level of sophistication in Logo programming was reached and many important language constructs were understood only at a superficial level. Lacking the deeper understanding required for meaningful programmming initiatives, many children tended to engage in trial-and-error activities accompanied by little planning or reflection (Leron, 1983). While such activities may be natural during the first exploratory stages of a new topic, their persistence over time may inhibit progress by ruling out the possibility for pre-planning and debugging.

The above research findings have caused some concern among educators. Particularly disturbing were the conclusions of the Bank Street College reports, stating bluntly that "the pedagogical fantasy... that Logo can serve as a stand alone center in classrooms for learning programming and thinking skills -- does not work" (Pea, 1983). Logo supporters disapproved of the methods used in the Bank Street studies, while some skeptics (particularly decision makers) quoted these studies, sometimes as a rationale to avoid spending time and/or money on setting up Logo programs; after all, here was "proof" that the Logo vision didn't work!

Evaluating "Logo" Vs. Evaluating Specific Logo Environments

Much of the dispute and confusion seem to have originated from different interpretations of the "piagetian", discovery oriented, "learning without curriculum" approach advocated by Papert. The myth that the teacher has little role to play in affecting children's learning in a Logo environment has been consistently perpetuated in most reports about children's Logo activities. Staring with the Brookline project (Papert, et al., 1978), other research studies have followed the same tradition of leaving out any description pertaining to instructional interventions, even though the children participating in these projects have often been exposed to a whole range of ideas, techniques and challenges by very competent teachers and educators. The Bank Street studies, while also following this general approach, have actually helped to dispel the above myth by showing that without substantial teacher support Logo did not produce the expected learning outcomes.

Recently, an increasing number of educators and researchers have commented on the crucial role the teacher plays in providing children with sufficient direction, guidance and support in their Logo programming. Teacher intervention, in varying degrees and forms, is now usually considered an integral part of the typical Logo environment and is the focus of several research studies (e.g., Noss, 1984, Hillel, 1984).

Besides teacher intervention strategies, there are a number of other variables in a Logo environment, such as teacher characteristics, teacher's level of experience in Logo, resources and curriculum materials available, classroom management routines and social climate, which affect the children's learning experience and hence the benefits they derive from the Logo program. All of these variables have to be taken into account in any serious evaluation study of Logo. Moreover, research efforts have to be directed towards examining and comparing various approaches to teaching Logo and different kinds of Logo environments in an attempt to determine the most effective approaches that will lead to maximum learning benefits. Long term developmental studies in which teachers and researchers collaborate in observing and documenting the various phenomena and processes that take place in typical Logo environments are needed. Recent Logo projects in this category include the second Brookline Logo project (Watt, 1982), the Chiltern Logo Project (Noss, 1984) and the recently completed Queen's University "Creative Uses" Project (Burnett, Higginson, et al., 1983) funded by the Ontario Ministry of Education.

Research Directions

Long term studies of the type described above, should help us gain the knowledge and insights necessary for determining effective approaches for implementing Logo environments, leading to the development of detailed teacher guidelines and resource materials for Logo programs for a variety of grade levels, populations and levels of experience with Logo.

Specific research topics dealing with the kind of knowledge required for reaching the above objectives include the following:

A. *Determine Prerequisite Cognitive Skills for Specific Levels of Programming.*
While the Logo philosophy implies leaving control and decision making in the hands of the learner, recent studies have reported recurring frustrations suffered by children who tended to choose goals above their ability level (Cohen, 1984). At the same time, even when teachers did intervene, they have occasionally over-estimated their students' level of readiness and consequently introduced new concepts prematurely or suggested projects that were too difficult (Burnett, et al., 1983). Several researchers have identified specific programming concepts (e.g., recursion, Kurland, & Pea, 1983, or procedure interface, Leron, 1983) that were beyond the reach of the children participating in the studies, even after a long term exposure to Logo programming.

In order to prevent needless (and mostly useless) frustrations, teachers (particularly those unexperienced in teaching Logo) should be made aware of expected performance levels and appropriate activities for different age groups and levels in Logo. There is a need to articulate intermediate stages of competence in Logo mastery and determine constraints on Logo programming performance in terms of prerequisite cognitive skills. Normative information about children's Logo programming ability at the various stages of learning Logo (within specific Logo environments and populations) should be established. Such information should prove indispensable in setting up Logo programs and developing resource materials.

B. *Establish Links Between Children's Ability Level in Logo and Their "Piagetian" Cognitive Development Stage.*
This need has been articulated by a number of researchers (Krasnor, & Mitterer, 1983, Pea, & Kurland, 1983) and has been recently mentioned in connection with an on-going debate as to whether young children should be exposed to "Logo before Lego" (Barnes, & Hill, 1983). The latter researchers contend that children in the pre-operational stage are not likely to benefit much from exposure to Logo, a view disputed by a number of pre-school educators. On the upper level of the cognitive development hierarchy, Krasnor and Mitterer (1983) express their view that transfer of

learning into other domains, particularly in the general problem solving heuristics area, is not likely to occur for children who have not reached the formal operational stage (around age 11-12). Clearly, all such statements are in conflict with Papert's view that substantial exposure to Logo may affect (and probably accelerate) the cognitive development of children (Papert, 1980). However, these issues are not easy to resolve and will require longitudinal studies.

C. *Study Typical Learning Processes and Cognitive Styles of Different Child Populations and Determine Effective Instructional Strategies for the Various Populations, Styles and Needs.*
Several studies have already addressed these questions for average children of particular age groups (e.g., Noss, 1984, Hillel, 1984).

The information gathered through extensive research efforts focussing on the questions discussed above will be of immediate practical value to both novice and experienced Logo teachers, to teacher trainers in pre-service and in-service training programs, as well as to school boards involved in planning and setting up Logo programs and curricula.

References

Barnes, B.J., & Hill, S. (May 1983). Should young children work with microcomputers -- Logo before Lego? The Computing Teacher, 11-14.

Burnett, J.D., Higginson, W.C., Carmichael, H.W., Moore, B., and Pollard, P.J. (1983). A Multisite Evaluation of the Creative Use of Microcomputers by Elementary School Children. Working Paper. Queen's University, Faculty of Education, Kingston, Ontario.

Cohen, R. (1984). Logo in the grade two classroom: A one year study. Department of Measurement, Evaluation, and Computer Applications, The Ontario Institute for Studies in Education, Toronto, Ontario.

Higginson, W. (June 1984). About that rose garden: Remarks on Logo, learning, children and schools. Pre-proceedings of the 1984 National Logo Conference, MIT, Cambridge, MA.

Hillel, J. Mathematical and programming concepts acquired by children, aged 8-9 in a restricted Logo environment, Research Report No. 1, Department of Mathematics, Concordia University, Montreal, Quebec.

Hillel, J., & Erlwanger, S. (1983). Observations, Reflections and Questions About Children's Logo Learning. Department of Mathematics, Concordia University, Montreal, Quebec. (Mimeo).

Howe, J.A.M. (1981). Learning Mathematics Through Logo Programming. Research Paper No. 153, Department of Artificial Intelligence, University of Edinburgh, Scotland.

Howe, J.A.M. (1980). Developmental stages in learning to program. In F. Klix & J. Hoffman (Eds.) Cognition and Memory: Interdisciplinary Research of Human Memory Activities. Amsterdam: North-Holland.

Krasnor, L.R., & Mitterer, J.O. (June 1984). Logo and the development of general problem-solving skills. Alberta Journal of Educational Research, 30, 133-144.

Kurland, D.M., & Pea, R.D. (1983). Children's Mental Models of Recursive Logo Programs. Technical Report No. 10, Center for Children & Technology, Bank Street College of Education, New York.

Leron, Uri. (1984). Some Problems in Children's LOGO Learning. University of Haifa, Israel. (Mimeo).

Noss, R. Explorations in mathematical thinking: Some implications from Logo classrooms. (June 1984). Pre-Proceedings of the 1984 National Logo Conference, M.I.T., Cambridge, MA.

Papert, Seymour. (1980). Mindstorms. New York: Basic Books.

Papert, S., Watt, D., diSessa, A., Weir, S. (1979). Final report of the Brookline Logo Project, parts II and III. Logo Memos 53 and 54, M.I.T., Cambridge, MA.

Pea, R.D. (June 1984). Symbol systems and thinking skills: Logo in context. In Pre-Proceedings of the 1984 National Logo Conference, M.I.T., Cambridge, MA.

Pea, Roy D. (1984). LOGO Programming and Problem Solving. Centre for Children and Technology, Bank Street College of Education, Technical Report No. 22, (12), New York.

Pea, R.D., & Kurland, M. (May 1983). On the cognitive and educational benefits of teaching children programing. New Ideas in Psychology, 11(5), 2-14.

Shultz, T.R., McGilly, C.A., Pratt, C.C., & Smith, J.S. (1984). The Effects of Learning Logo on Children's Logical and Mathematical Thinking. McGill University, Montreal, Quebec.

Solomon, C. (August 1982). Introducing Logo to children. BYTE, 7(8).

Robotics With Logo And A Tasman Turtle

Dave Goforth
Science North
100 Ramsey Lake Road
Sudbury, Ontario P3E 4S8

LOGO and a Tasman Turtle together can be used to investigate theoretical and practical issues in robotics. A series of exercises is described which requires minimal LOGO skills but illustrates the concepts of sensory input, action and internal representation of environment. It also brings home the fundamental importance of real-time issues and mechanical considerations without once requiring the use of a screwdriver.

Science North is a participatory science centre in Sudbury, Ontario catering to regional residents and visitors alike. Local citizens are encouraged to engage in projects which might entail several trips to Science North and enrichment teachers in the Sudbury area have brought classes to the centre on this basis. Typically, students participate one afternoon a month though many devote an additional four hours per week of their own time as junior science associates.

In the computer exhibit area, the infrequency of attendance precludes a strongly sequential program. Instead, I have given one introductory session on LOGO then provided a sequence of microworlds to investigate, assuming that gifted students in grade 6 could pick up programming skills as required in their pursuit of domain knowledge. Of course, those attending on their own time have the opportunity to hone their LOGO skills.

The domain of interest is basic robotic principles. Before discussing the microworlds, I will briefly describe the significant features of the equipment involved which consists of an Apple IIe with disk drive and monitor, a Tasman Turtle and LOGO software. The Turtle is a physical analog of the screen turtle. It is hemispheric in shape with a wheel in each side and knobs front and rear for balance. Each wheel is driven by a stepper motor. When both wheels turn in the same direction, the Turtle moves in a straight line, either forward or backward; when they turn in opposite directions, the Turtle rotates left or right about its centre. Commands from the computer activate the motors. At the centre is a bracket for a marker which can be raised or lowered in PU/PD fashion.

Unlike the screen version, the Turtle has a bumper ring around its perimeter. Inside the ring, located at the front, the rear and each side are sensors that are activated when the ring is pushed against them. Signals from the sensors can be read by the computer.

The primary aim of the exercises with the Turtle is to investigate robotics programming in terms of sensory input, action, and internal representation of the

BEST COPY AVAILABLE

environment. This aim is addressed according to the accepted LOGO philosophy of providing a set of primitives defining a microworld. The learner gains intuition about the microworld concepts by programming with the primitives which presumably embody the concepts in pure form.

The secondary aims are to improve programming skills and to confront real world problems with time and mechanical devices. To accomplish these aims, the same microworlds are used but the primitives themselves become the objects of focus. For a real world Turtle, the primitives are impure in the sense that they do not accomplish exactly what they are supposed to. Controlling this random behavior involves opening up and adjusting the definitions of the primitives.

The remainder of this paper describes the microworlds, their purposes and suggested activities.

1. Just like the screen? A set of primitives to control the Tasman Turtle is provided. These are analogous to FD, BK, LT, RT, PU, PD. Drawing with them discloses that the angles aren't correct. LT and RT must be opened and calibrated. (Perhaps the marker bracket must also be bent so the Turtle doesn't draw an arc while rotating.) Investigation shows that a sequence of small advances or turns is less reliable than a single action of the same cumulative magnitude; stopping and starting should be kept to a minimum. Skidding caused by bumping into something is disasterous.

2. Interactive control. A set of dynamic primitives is provided in the form of single key stroke commands. These initiate action (moving or turning) which continues until interrupted by another command. These commands are to be revised so that the action will stop when the Turtle bumps something as well as when another key is pushed. At the top level, we start to see sensory input feeding back to control action. Implementing the feature gets deeper into how the primitives are defined. For each step of action, signals must be sent to both wheels. If the signals are sent too frequently, some are lost and the Turtle's motion is erratic. (So that's why the dummy repeat loop was in the commands in exercise 1.) If the commands are sent too infrequently, the Turtle moves jerkily because it is, in fact, starting and stopping. In this exercise, the polling of the sensors and of the keyboard between transmissions to the wheels takes too long if not efficiently coded.

3. Blind Turtle. The primitives in exercise 2 are reused. They are revised so that the action terminates either when something is bumped or when a counted number of steps has been made. Keyboard interruption is removed because the Turtle is to move independently. Her area of movement is restricted by an encircling wall of blocks and she is to be programmed to feel her way around the enclosure. The follow the left wall strategy is fairly obvious but implementing it with the crude sensors and actions of the Turtle is not. (The algorithm is left as a challenge to the reader.)

4. Walk a Mile in my Shoes. Again the primitives of exercise 2 are used (actions which are interruptable by the keyboard or the sensors) and the enclosing wall of blocks is retained. First an addition is

BEST COPY AVAILABLE

made to the code so that, when the Turtle bumps something, a graphic display on the screen shows a circle representing the Turtle and a flashing point representing the sensor(s) which sent the signal. Now the line of sight to the Turtle from the position of the computer is obstructed and the humans attempt to 'play Turtle' by steering her around the enclosure guided only by the graphic display. The frustration of not being able to look at the Turtle makes clear the local/global distinction. At this point we stop for a little fun on a flight simulator, rationalized as further investigation of the local (view out the windscreen) versus global (instrumentation) viewpoint.

5. Cartographer This is the culmination of the sequence. First the primitives of exercises 2 and 3 are combined so that an action may be interrupted by the keyboard, by the sensors or by a counted number of steps - a final, tricky real time challenge. A new keystroke command is added to initiate the Turtle's blind wall-feeling algorithm. From the keyboard, one can now take control of the Turtle and direct her actions or relinquish control to the wall feeling strategy. Now we want to get the Turtle to do for us at the keyboard what the instruments do for the pilot in the airplane: provide the global view. The graphic, of the circle and the flashing points, is to be replaced by a map of the enclosure. All that is required is to scale and display the Turtle's world movement as screen turtle movement. When control is relinquished to the wall-feeling algorithm, a map is created, an internal representation of the external environment. Of course, all the issues of error and calibration arise again. Clearly, the final exercise is not the work of one afternoon; it is however, a worthwhile undertaking. The mapping Turtle is now on the floor of Science North as an exhibit.

The next logical step would be to have the Turtle generate an internal map, not a graphic display, which she could use herself. At that point, it would be possible to start modelling goal directed activity in the enclosure such as searching for an object.

Bibliography

-- "Tasman Turtle Information Booklet", Harvard Associates, Somerville MA, 1983.
(no author accredited)

BEST COPY AVAILABLE

SOFTWARE

Alcohol Education: The Choice Is Yours

Barbara S. Thomas, Ph.D.
University of Iowa
College of Nursing
Iowa City, Iowa 52242

Alcohol Education: The Choice is Yours is designed for school children and professionals who work with them. Its 5 sections are: (1) Facts About Drinking: A Self-Test, (2) Your Interests Please?, (3) Alcohol's Effects, (4) Taking Chances: A Game, and (5) Key Terms and Definitions. Users are encouraged to start with the self-test to identify their misconceptions. Section 2 lets them type in questions anonymously for later discussion. The tutorial uses graphics heavily to illustrate physiological and emotional effects of alcohol. Taking Chances confronts users with choices about drinking. The 5th part is a glossary of key terms.

There is a growing consensus that primary prevention is the only feasible way to deal with the nation's number one drug problem—alcohol abuse. Studies have shown that a focus on alcoholism and use of scare tactics are not effective. Such an approach is both unrealistic and psychologically unsound for several reasons. It fails to acknowledge that alcohol use continues to be an important part of life; it has been for many years. It fails to respect the different values and beliefs that children learn in their homes. And most importantly, it conflicts with students' personal observations and experiences. Many who have experimented with alcohol or have been allowed to use it at home, promptly discount horror tales about alcohol's immediate disastrous effects. The administration of a "dose of prevention" via a film, interview with a recovered alcoholic or law enforcement officer has also been shown to have little effect on the subsequent drinking behavior of adolescents or adults.

Education about alcohol use or non-use must be addressed as a facet of responsible decision-making. The importance of peer pressure in promoting experimentation and use of beverage alcohol has been recognized for a number of years. Children need opportunities to confront and rehearse situations in which they will experience peer pressure, recognize it for what it is, and make choices based on sound information. Recent studies have shown that alcohol education programs which use realistic, objective information, provide models for responsible decision-making, and take into account the complex interactions among peer pressures, home influences and the media, can have a positive impact as a primary prevention strategy. To date there have been no published reports of the use of computer-based instructional (CAI) materials for primary prevention of substance abuse. This work, which is part of a larger project, Using CAI for Primary Prevention of Substance Abuse, was supported by a Special Projects Grant from the Division of Nursing (DHHS).

The rationale for choosing the microcomputer as a primary prevention tool is as follows:

This Work was supported by means of a three-year special projects grant (Number 1D10NU270075) from the Division of Nursing, DHHS.

1. Learning experiences should build upon students' prior learning. The computer is ideal as an anonymous, confidential, non-judgmental aid in assessing user's knowledge and attitudes.

2. Up-to-date information about alcohol can be presented in a self-paced mode based on individual's interests and needs.

3. Integration of alcohol education with other school subjects is preferred over separate coursework. The use of CAI facilitates such integration with health topics, history, family living, consumer education, etc.

4. Although curriculum guides and learning materials are widely available, many teachers do not feel comfortable with the subject and fail to use these materials. The CAI materials provide professionals with correct, effective programs to up-date their own learning and to use with students.

5. Special CAI programs for professionals provide them with skills they need. A CAI program, "Inoculation Theory and Substance Abuse", teaches users ways to identify children at greatest risk and strategies to lessen the probability that such children will become substance abusers.

6. The CAI programs are based upon a team approach. A CAI program "Team Building" focuses on goals, roles and procedures for collaborative approaches to primary prevention. This program is expected to enhance cooperation among teachers, counselors, school nurses and parents.

The objectives for the program are stated explicitly in the manual and the program itself as abilities to:

1. describe alcohol use in the history of man, listing differences as functions of time and cultures
2. list important facts about beverage alcohol and its effects
3. identify human needs (motivations) which are associated with the use of alcohol
4. identify alternative ways of meeting such needs
5. identify physiological, psychological, social and emotional problems associated with irresponsible use of alcohol
6. analyze media messages which promote use of alcohol
7. identify effects of alcohol abuse on health, family, friends, work and safety
8. adopt a logical decision-making process of examining alternatives for meeting needs
9. identify responsible coping skills for dealing with pressure to misuse or abuse alcohol
10. explain the role of laws, rules and regulations in preventing or penalizing substance abuse

The program is menu-driven and consists of five sections. At the beginning of the program, the user is given directions for making choices, continuing the program, returning to the menu and exiting the program. Sections can be repeated as often as necessary.

The first part is a brief self-test, "Facts About Drinking". The items focus on common misconceptions that people have about drinking. Users might know the information, miss the item or guess. Regardless of their responses, they are given correct information as feedback along with the "Right" or "Wrong".

The second section, "Your Interests Please?" invites users to type in questions they may have about drinking. Anonymity encourages children to ask questions they may be reluctant to raise in a group. The teacher, nurse or counselor can use a variety of ways to handle the questions which they can garner from the diskette. Another advantage of this approach is that the user who "hustles" the leader by entering capricious questions or comments can be ignored.

The third program, "Alcohol's Effects", is a tutorial. Graphics are used very heavily to illustrate physiological and psychological effects of the drug. The potential dangers of using alcohol with other drugs are explained. The fact that alcohol is a depressant is clarified and stressed. The concept Blood Alcohol Content (BAC) is explained. Impairments caused by different levels of BAC are described and summarized, including the legal definition of driving while intoxicated. Needs which give rise to drinking and alternative ways of meeting those needs are described. The tutorial includes questions which the user must answer throughout the lesson.

Taking Chances" is a game which can be played by two users or one user against the computer. Twenty-four situations are arranged in three columns depending on how difficult the situation is, i.e., the probability of getting a right answer. The five point questions have three correct responses and one incorrect response hidden behind the four doors, labeled A,B,C and D. The ten point questions have two correct and two incorrect responses, and the fifteen point doors hide only one correct response and three incorrect answers. A six part stick man represents each player; a body part disappears every time an irresponsible choice is made. The game ends when one player gets fifty points or loses his body parts. The situations are generated randomly so that repeated games will be different, and success cannot depend on memory.

The last section, "Key Terms and Definitions", simply provides the user with a list of terms. He or she can choose any or all to be defined. After each definition, the user is returned to the list of terms.

Formative evaluation was a continuous process throughout the development of the program. Programmer-author review was supplemented with assessments by colleagues, elementary school teachers, school nurses, elementary, secondary and college students and parents. Revision were made based on these evaluations, and the program was placed in schools of nursing, public schools, private schools, and hospitals for field testing. The results of these tests are not complete at this time, but will be available at the AEDS/ECCO Conference. It is expected that evaluations from over 1000 users will be reported.

The real test of any educational program designed for primary prevention is its efficacy in preventing the targeted behavior. Application for further support to assess the effectiveness of the program is planned. The complete series of CAI programs, together with the manual and especially designed further activities will be evaluated using a pre-test/post-test experimental design. Both knowledge and attitudes will be measured. However, attitudes do not predict behavior very precisely, and long-term as well as short-term effects should be examined. Consequently the onset, frequency of drug use and incidence of drug abuse will be investigated over a five year period, following use of the programs by elementary students.

It is hypothesized that the use of this program in grades 4-6 will be effective and sufficiently robust to result in significant differences between the two groups of children in the evaluation study. The series consists of four programs for joint use by professionals and children:

I. Drinking and Not Drinking: The Choice is Yours

II. Keep Off The Grass

III. Six Classes of Psychoactive Drugs

IV. Techniques of Persuasion

Program II deals with marijuana. Program III deals with amphetamines, barbiturates, cocaine, opiates, hallucinogens and volatile nhalants. "Techniques of Persuasion", consists of a tutorial, two simulations, a self-test and a game.

The programs designed for the professionals' use are as follows:

I. Inoculation Theory and Substance Abuse
II. Team Building

An extensive section in the manual accompanies the first program. It provides the user with a review of the literature concerning factors related to substance abuse, its etiology and correlates. Ways of improving children's impulse control, critical thinking skills and self-worth will be stressed in the CAI program.

The team-building program is based on the work done by Rubin, Plovnick and Fry (1975) in helping health professionals work more effectively together. The basis for this approach is in the organizational behavior guidelines which lead to a task-oriented approach for collaboration. If differences among individual's perceptions of goals, roles and procedures are addressed, the difficulties attributed to interpersonal problems disappear or are ameliorated.

The program on alcohol education will be demonstrated, and additional information about the entire series will be reported in the paper.

References

Rubin, I.; Plovnick, M.; and Fry, R., Improving the Coordination of Care: A Program for Health Team Development, Cambridge: Ballinger Press, 1975.

Thomas, Barbara and McClain, Don, Use of Microcomputers in Continuing Nursing Education, DHHS Grant Application, 1982.

Use Of CAI To Improve Instruction In Nursing Research

Barbara S. Thomas, Ph.D.
University of Iowa
College of Nursing
Iowa City, Iowa 52242

The wide variety of teaching strategies which have been reported for research preparation in nursing education have been developed almost exclusively for local use. No computer-based instructional materials were available in 1984. This paper describes the development of 10 computer-assisted-instruction (CAI) programs, designed for use at both undergraduate and graduate levels. All programs contain tutorials, problem-solving activities and self-tests. Four simulations provide users with opportunities for decision-making and observing the impact of selections on outcomes, assumptions, limitations and costs. Manuals containing guidelines accompany each volume.

Introduction

Although research preparation is seen as a necessary component of nursing programs there has been a curious dearth of audiovisual or technology-based aids to augment teaching research methods. Announcements of new videotapes, filmstrips or other products to aid students' learning about theoretical or clinical aspects of nursing are received daily. But products designed for instruction in nursing research have been limited solely to textbooks.

Nursing faculty have been active however, in developing instructional materials for teaching research. A number of interesting strategies have been reported in the literature and at conferences for nurse educators. A bibliography of articles about teaching nursing research, 1978-1982, was compiled by Thomas and Price (1982) . Although 36 articles were located, none reported the use of computer-based instructional materials (Thomas & Price, 1982).

At the conference, "Methods of Teaching Nursing Research", 27 reports described diverse topics (Thomas, 1982) but only 3 pertained to CAI.

Thus there have been spotty efforts but no systematic attempt to develop CAI or assess its effectiveness in research preparation. This report describes the development of two volumes of programs, each consisting of five programs. Formative and summative evaluation processes involving faculty and students are also reported.

Program Development

At The University of Iowa, CAI laboratory staff announced a competition for all faculty. A call for proposals was issued for projects designed to use microcomputers in teaching. The award consisted of technical assistance, supplies, use of hardware and assistance in evaluating and refining the products. My proposal, "Using Microcomputers to Improve Instruction in Nursing Research" was selected. This proposal described examples of programs expected to be useful in both the undergraduate and graduate programs, stating that at least eight programs would be completed, evaluated, refined and disseminated. When complete, Nursing Research CAI consisted of 10 menu-driven programs divided into two volumes with versions for either APPLE or IMB-PC microcomputers. A manual was prepared for each volume.

A team approach to program development was used for each phase of the process. Observing novices' use of programs was also an essential step in locating confusing or ineffective components of lessons as well as bugs. Revisions and refinements were made, based on this information. For each section in a CAI program, the question was asked, is this material uniquely suited to the CAI mode? The development-review-revision loop was repeated until users reported no additional flaws or difficulties in using each program.

For summative evaluation, faculty colleagues and both undergraduate and graduate students used the programs and completed anonymous evaluations for each one. The CONDUIT form (1972) was used for faculty appraisals; students' evaluations were less detailed. The revisions which resulted from this phase of program development concerned program operation more than content.

The CAI Programs

Overview of the Programs

Program development paralleled the research process. Volume I consists of five programs about the early phases: (1) Research Terminology, (2) A Systematic Approach to Critiquing, (3) Tables and Graphs, (4) Formulating Researchable Problems, and (5) Sampling. Volume II consists of the following: (1) Data Collection I: Criteria for Judging Data Collection Instruments, (2) Data Collection II: Interviews and Questionnaires, (3) Data Collection III: Other Methods of Data Collection, (4) Analyzing Qualitative Data, and (5) Analyzing Quantitative Data.

Each program is menu-driven and contains tutorials, problem-solving activities, and self-tests. Some contain drill and practice exercises; others contain simulations and graphics. Students must enter an ID number to take each test and their performance is recorded on a separate diskette designed for instructor retrieval of data. When scores for one class have been recorded, the diskette can be erased, ready for another group's use.

Volume I

Research Terminology, the first program completed, consists of a tutorial about twelve key concepts and a self-test. Questions requiring users to apply the concepts are interspersed throughout the tutorial to reinforce learning and to keep the user actively involved.

A Systematic Approach to Critiquing is based on mastery learning to be demonstrated by applying guidelines from the tutorial to one, two or three self-tests. The tutorial provides guidance for assessing the purpose of the study, the review of the literature, the precision and significance of the problem statement , the methodology, and results and conclusions. All four articles for the tutorial and self-tests focus on substance abuse. The student must read the first article before signing on to do the tutorial. At the end of the lesson, the student is given feedback about her or his performance in the tutorial, and a list of three publications to make a selection for the subsequent self-test. After reading the selected article, the student signs on for the self-test. A score of 80% or better is required; if a lower score is recorded, the user is given a remedial reading assignment and a choice of the other two articles for another self-test. Failure (<80%) on the second self-test results in more reading and another test.

The program, Tables and Graphs, consists of a tutorial and self-test on tables and a tutorial and self-test on graphs. Users are given practice in interpreting and constructing tables and graphs. Graphics and simulations play important roles in explaining line, bar and pie graphs. At the end of each section, the guidelines for constructing tables and graphs are illustrated and summarized.

Formulating Researchable Problems consists of the following menu:

A. Identifying the Problem
B. Reviewing the Literature
C. Formulating Researchable Problems: A Simulation
D. Self-test
E. End Program

The first section is a tutorial which describes sources of research questions and illustrates the process of identifying and refining researchable problems. At two points, the user is given an idea (stimulus) and asked to type in a research question. Instructors can print the classes' collections of ideas for discussion or to be posted. Nursing students can review each others' ideas and see for themselves that a theory or research report can stimulate a wide variety of research questions. The section, "Reviewing the Literature" is also a tutorial. Four reasons for reviewing the literature are explained. Preparating a review based on topics rather than authors is emphasized. Practical issues are described as well.

The simulation uses the topic, "Recovery from Hysterectomy," to approximate the real process of identifying a researchable problem from a literature review. Ten abstracts are available; users can choose any five of these based on the titles, authors and names and dates of the periodicals. The computer prints out the chosen abstracts and helps the user to organize the information by topic. using this mini- review of the literature, each user formulates a researchable problem. The differences resulting from different combinations of five articles demonstrate the individual and cumulative nature of research, and also the consequences of an inadequate review of the literature. Instructors may print out the problems for the entire class for discussion or posting. The self-test consists of 10 short answer questions. If two or more questions are missed, the student is instructed to repeat the tutorials.

The Sampling program begins with a section on basic terms and definitions because learning the vocabulary of sampling is more than half the battle. The next section of the lesson is a tutorial, in which the types of sampling, their requirements and advantages are explained. Adequacy and representativeness are stressed and are described as mutually dependent criteria. Simulations provide students with experience in drawing samples.

Volume II

Data Collection I: Criteria for Judging Data Collection Instruments contains two tutorials to stress the importance of validity and reliability, explaining different ways of establishing each. Additional factors, e.g. efficiency, sensitivity, reactivity, range and objectivity are treated as "Other Criteria" in the third tutorial. Each concept is explained, and examples from nursing research are used to illustrate ways of establishing the different requirements.

Data Collection II: Interviews and Questionnaires has a menu made up of three tutorials, a simulation and a self-test. The tutorials, "Interviews", "Questionnaires" and "Comparison of Interviews & Questionnaires", outline advantages and disadvantages of each and compare the two approaches. The research process is treated as a series of decisions to illustrate the fact that subsequent choices are limited once a series of decisions has been made. In the simulation, the user can choose to collect data for four variables by either interviews or questionnaires. Additional choices within each of these approaches are permitted. At the end of each simulation, the computer becomes a reporter to describe results. Users selecting interviews for data collection will learn the impact of their decisions about ways of recording the data. For example, use of a tape recorder will result in the most complete, accurate dataset, but costs for transcription and also reduction of sample size because of potential subjects' reluctance to be tape recorded necessarily follows. Similarly, decisions about the format and form of questionnaire items affect limitations and assumptions as well as results. At the end of the simulation, users are given realistic feedback about both theoretical and practical considerations. Moreover, they may repeat the simulation making different choices to learn about their effects.

Data Collection III: Other Methods of Collecting Data is a very large program. It consists of eight tutorials on (1) Physiological and Biophysical Measures, (2) Psychosocial Measures, (3) Records, (4) Observation, (5) Critical Incidents, (6) Projective Techniques, (7) The Delphi Technique and (8) Simulations. Each tutorial contains guidelines, illustrations and exercises requiring application of the principles.

Qualitative Data Analysis consists of four sections: (1) Content Analysis: Definitions and Examples, (2) Guidelines for Data Reduction, (3) Attitudes Towards Research: A Simulation, and (4) Self Test. The first two sections are tutorials which explain content analysis for semantic or manifest content or for feeling tone. Although guidelines are provided to promote use of a systematic approach, the inevitable subjectivity introduced in qualitative data analysis is explained and illustrated. The simulation requires application of the principles and guidelines to a small set of data concerning the attitudes toward research.

Quantitative Data Analysis consists of three tutorials, information sources for further study, and a self-test. The first tutorial, "Definitions and Decisions", portrays the research process as a series of decisions pertaining to WHEN? WHO? and WHAT? Early planning is stressed and users are reminded that consultation with statisticians or computer specialists should be initiated before data collection begins. An extensive application of the guidelines is provided by a simulated study on middle-aged women's exercise programs. The second tutorial, "Data Management and Preparation", is a highly practical guide for maintaining accurate, confidential datasets efficiently. Preparation for computer analysis of data is explained. The third tutorial, "Computer Analysis of Data" describes the use of mainframe, mini and microcomputers in data analysis. The use of statistical packages is described with its most troublesome part, writing format statements, explained and illustrated. The fourth section is a bibliography divided into two parts: "Data Analysis in Nursing" and "Use of Computer-based Statistical Programs".

Using the Programs

Since the research process formed the basis for developing Nursing Research CAI, the programs and manual are independent of any textbook. However, reading assignments in major nursing research textbooks are included in the manuals for each program. The programs may be used to introduce topics, reinforce prior learning or for review. They are all designed for independent, self-paced learning. Students can repeat sections or entire programs as desired. Instructors can garner the self-test scores for evaluating students' progress or they can simply suggest that their students test their own learning. In either event, the user is given correct information as feedback for each question so that the self-tests serve as learning experiences as well as evaluation tools.

REFERENCES

Thomas, Barbara and Marjorie Price, Evaluating Research Preparation in Baccalaureate Nursing Programs, Iowa City: University of Iowa, 1982.

Thomas, Barbara, Methods of Teaching Nursing Research, Iowa City: University of Iowa, 1982.

Marketing Educational Software: A Publisher's Perspective

Janet Paine
Holt, Rinehart and Winston of Canada
Toronto, Ontario

This overview is designed to provide educational software developers with a sense of the market for which they are creating. It discusses the structure and dynamics of the industry in terms of its marketing and distribution functions. Author/publisher responsibilities and remuneration policies are considered, along with publishing criteria for selecting software. Comment on future industry trends is also provided.

In recent years educational publishers have made a considerable commitment to the publishing of educational software. It has become a crucial element in successful textbook packages, and its unique potential as a dynamic educational tool has resulted in a dramatic restructuring of marketing strategies in educational publishing.

There are several methods available to publishers for acquiring software. One way is to acquire an already developed product and introduce it into their educational market. THE BANK STREET WRITER has been adopted by Scholastic Publishing in this manner and Mindflite, a Canadian software house, uses a similar distribution process.

Another avenue which publishers take is to contract and provide support for a software designer who has a worthwhile project that would benefit from developmental assistance. At Holt, Rinehart, and Winston of Canada, special provision has been made in the form of developmental grants to encourage designers to create new and innovative software.

Some publishers make use of an in-house approach. This involves hiring or contracting an instructional design team who works closely with editors and marketing personnel to develop a product specifically tailored for integration with basal curriculum programs. This approach is used extensively in our school division at Holt, Rinehart and Winston where the software we are developing is to be used as an ancillary complement to cross-curriculum textbook packages. THE VOYAGE OF THE MIMI was developed in conjunction with Bank Street College in this manner, and our basal Language and Math Series will employ it as well.

No one approach is necessarily used to the exclusion of others, and sometimes a combination of approaches works best. Usually directional decisions are based on an assessment of a variety of factors, including budget allowances, the need for quality control, publishing deadlines, and editorial considerations. Most important is a preliminary analysis to determine specific market needs. Holt, Rinehart and Winston's recent survey of the software requirements of Economics, Accounting and English instructors at the college level is an example of this prerequisite ground work.

The most important criteria for selecting educational software are high program quality and overall educational merit. The concept also should embody specific educational goals and meet pre-defined pedagogical standards. Open-ended and creative tool applications, problem-solving simulations and games are particularly in demand. The material should be innovative as well as easy to use. You should design for machines with a broad distribution base and write in a programming language that is easily transportable.

Many educational publishers discourage the development of stand-alone software. "Stand-alones" have done poorly in the educational software market, primarily because teachers lack the time and training to integrate them into their existing curriculum. Our market research at Holt Canada suggests that teachers prefer software which can be used with existing texts and ancillary material.

Once the conceptual directions of the software have been consolidated, the detailed specifications for the final design can be prepared. These will include design criteria, the specific scope of the project in terms of hardware requirements, coding language, content, anticipated costs, and estimates of how long it will take to complete. Be realistic in estimating production team requirements. At a minimum a production team will include a co-ordinator, an instructional designer, a content expert, an art director, assembly and packaging people.

Your software package should be geared to the specific needs of a fairly large group of students. Wherever possible the completed software should be "beta-tested" with a group similar to the intended audience. The credibility of software is greatly enhanced if it can be demonstrated that it is effective in an educational environment similar to the one in which it will be marketed.

Teachers who have tested a product are usually willing to report on its effectiveness and its limitations. Keep in touch with the needs of teachers and other educators, especially those who are using software in their classrooms. This will help to avoid duplicating existing software packages and assure that the product you are designing is both current in its technology and relevant in its pedagogy. Do not be afraid to admit that the program may have bugs or limitations. It is only through recognition of these that a product can be improved.

For a publisher to even consider a software proposal he has to have a pretty good understanding of what you are planning to do. Take the time to present your concept effectively. Here are some suggestions. Use a combination of descriptive overviews, flowcharts and storyboards. Whenever possible provide disks containing sample programs. Outline how you see the project being implemented in an educational environment. Describe how it is valuable as an educational support tool.

Be sure to tell the publisher why the product is worth developing and marketing. Most publishers have guidelines that outline the information that they need to make an evaluation of a product's developmental potential. Ask for them and follow them.

Remember, software must be designed to enhance a student's comprehension of basic principles and to provide a hands-on opportunity to use the computer as a practical tool for implementing conceptual problem solving skills. It must also provide teachers with both the motivation and the methodology to use computerized learning aids effectively in their classroom.

Here is a model for an integrated software package. Disks should contain about six to eight programs each, selected from major topics where the content material lends itself profitably to computerization. The pedagogical approach should be one which links theory and application. It should be stimulating and challenging, and, as well, it should incorporate generally held principles of good software design. Avoid designing computer activities which can be implemented with equal success using traditional methods. A data disk which supports the lessonware can also be an attractive component of educational software. It extends the scope and range of the student's application disk, and, furthermore, it provides teachers with a means of personalizing the computer package to a class's specific needs.

A User's Guide should be accompany the package to provide instructions for running the software. This frequently is incorporated into the Teacher's Manual or into a Student Workbook. The workbook can be designed either to provide support material exclusively for the software, or as an integrated workbook that links the software to a specific textbook. A good in-depth description of an integrated curriculum package is provided by Lloyd W. Black in "Educational Software Development and the Integrated Curriculum: The Courseware Concept" which may be requested from the educational software editor at Holt, Rinehart, and Winston of Canada.

To help you in deciding on a marketable project, inquire into a publisher's needs and structure your proposal accordingly. Most publishers will be glad to provide you with a short list of the kind of software packages that they are looking for.

In selecting the machine for which you are going to develop your software be sure to choose one with a wide enough market base to insure a viable commercial market. You will also need to make provisions for maintaining and upgrading the courseware. It is important to establish who has responsibility for software maintenance, you or your publisher. Providing ongoing technical support is essential to any software product's credibility and most publishers will want a commitment to this effect.

Any publisher who decides to take on a software package is assuming considerable entreprenural risk. He is committed both financially and in terms of developmental time. Marketing costs such as advertising and promotion are his greatest expense. Unless your product meets both his standards and his marketing needs there is little incentive for him to assist in a product's development.

There are several ways in which publishers arrange to compensate their software developers. They may buy an entire package and retain exclusive rights to it, or they may encourage development in the form of developmental grants. They may license the distribution rights, or pay royalties on the finished product.

The method of compensation depends largely on the degree of the publisher's involvement in the product's development, and also how he envisions marketing it. Most common is the royalty arrangement.

Frequently, educational software designers have unrealistic expectations as to the profits they will see from their software. It is important that one get a clear picture of the potentials and limitations of anticipated remuneration.

In the past, one of the major limitations in marketing educational software was the lack of a well-defined and standardized machine base. This made it difficult to market the "stand-alone" programs oriented to individualized learning that were among the first products of educational technology. This type of package is still highly successful in training environments, but teachers in the high schools and college resisted what was frequently perceived to be a "teacher replacement device".

Resistance to software pedagogy was not the only reason that teachers were intially hesitant about "stand-alones". Typically, these packages consisted of multiple disk units priced anywhere from 200 to 500 dollars. The cost of such packages were prohibitive to many institutions.

Another area where educational software met resistance was with respect to its lack of design sophistication. Too often it was produced by novices who had just begun to learn the nuances of designing software. Much of the educational software produced lacked the attractive graphic component of commercial packages; and, worst of all, a large percent were pedagogically unsound.

An additional problem was getting the product to the user. Traditionally, educational publishers employ sales representatives to call on school administrators and college professors to present their books. Frequently these sales people felt that it wasn't worth their trouble to promote it.

Publishers overcame these marketing limitations in a variety of ways. New pricing strategies were adopted, demonstration software was developed. Integrated courseware packages were introduced such as "softbooks" which linked software with texts. Textbook sales were stimulated by the inclusion of these expanded software ancillaries, and sales reps have begun to view software as an incentive to sales, rather than an unproven commodity competing for their time. Finally, and perhaps most importantly, publishers themselves realized that by integrating software into existing text-oriented curriculums teachers could become acquainted with its potential in an content domain with which they were familiar.

The "integrated curriculum" approach has been tremendously effective as a way of introducing software to the college and secondary school markets in both the United States and Canada. Charles Parker's best-selling UNDERSTANDING COMPUTERS is an outstanding example of this at the college level, while the cross-the-curriculum project, THE VOYAGE OF THE MIMI developed in conjunction with The Bank Street School demonstrates the effectiveness of this approach in a school environment.

Recent studies on effective software marketing indicate that selling strategies must be undertaken from the time of a project's inception. Advertising and distribution methods should be determined right from the beginning.

Educational publishers use a variety of ways to distribute software. They advertise in catalogues and journals, they employ sales representatives, they exhibit at educational conferences, they offer educational seminars and they display their products in bookstores and computer stores. The choice of marketing method depends to a large extent on the kind of software being produced, and the market for which it is intended.

In spite of considerable market instability in the past, educational software shows every indication of settling into a period of settled and unprecedented growth. But developers are going to have to keep up with changes in technology if they are to produce software that is worth marketing. One area that educational publishers are exploring is that of the application package: word-processors, organizers, databases and spreadsheets. Developing educational packages that provide instruction in these areas is crucial if students are to be prepared for the most recent changes in computer technology. It is also becoming increasingly important to take advantage of the sophisticated visual and graphic capabilities now available to courseware designers. And finally, developers must be aware of and take advantage of new developments in authoring tools which are constantly expanding the possibilities for enhancing software design.

Educational software technologists will need to use all of these technological developments if they are to produce effectual and competitive product to meet the expanding needs of the educational market.

REFERENCES

Bank Street School. THE VOYAGE OF THE MIMI, Holt, Rinehart and Winston: New York,1985.

Billings, Karen et. al. "Courseware Development from the Publisher's Perspective, AEDS MONITOR, Sept./Oct. 1984, pp.6-13.

Black, Lloyd W. "Educational Software Development and the Integrated Curriculum: The Courseware Concept", Saunders College Publishing (in-house publication) 1984.

Parker, C. S. UNDERSTANDING COMPUTERS AND DATA PROCESSING, Holt, Rinehart and Winston: New York and Toronto, 1984.

Sigel, Efrem & Giglio, Louis. A GUIDE TO SOFTWARE PUBLISHING, Knowledge Industry Publications, Inc.: New York. 1984.

Electronic Thinking Cap: Microcomputer Enhanced Creative Problem Solving for Visually Oriented Students

Susan E. Metros
Department of Art University of Tennessee
Knoxville, Tennessee

This paper describes research being conducted on computer enhancement of creative problem solving for art students studying graphic design. The difficulty in teaching conventional creative problem solving is discussed, along with the benefits of computer enhancement. The author's model, "Electronic Thinking Cap," is presented and explained.

BACKGROUND

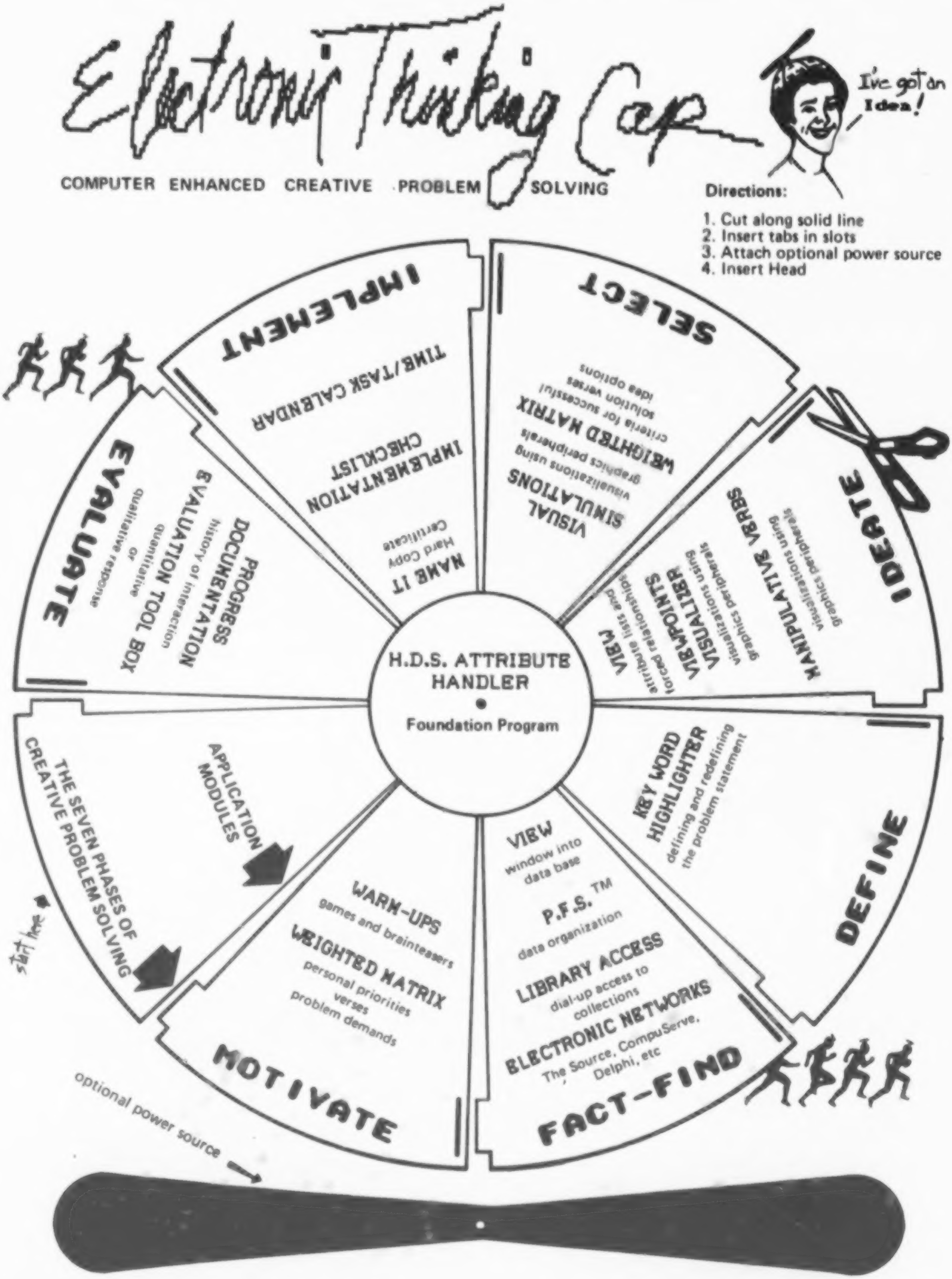
From the turn of the century until the 1950's, graphic design was referred to as commerical art. A commercial artist's job was to "decorate" a client's concept or idea. This is no longer the case. American graphic design has undergone an evolution (Fahnstrom, 1983). This evolution can be attributed to what Megatrends author, John Naisbitt reports as a megashift from an industrial to an information society (Naisbitt, 1984). This proliferation of information increases the need to interpret verbal and written words into an easy-to-understand visual format. The graphic designer has become this translator.

The graphic designer has emerged to become a primary contributor to modern visual culture. Today's designer plays a multi-faceted role, focusing on design as a process requiring multi-disciplinary skills. The new designer is a businessperson, a team player, a sociologist, a technician, a communicator, an artist, and, most importantly, a creative problem solver. This evolution of the design profession has forced the designer into an important decision making role. The design process is no longer client dictated. Today, a designer is hired with the expectation to carry a project through concept to final piece. The stages in this design process are as follows: defining and researching the problem, choosing criteria for a successful solution, generating a quantity of idea options, selecting the best of these options, producing the piece, and monitoring the piece's success in the market. This design process is the visual counterpart to a conventional creative problem solving model.

CONVENTIONAL CREATIVE PROBLEM SOLVING

It is the responsibility of design education to educate students in the analytical and process phases of graphic design. This kind of knowledge prepares them for a competitive job market that demands innovation. There are many models developed to stimulate creative thinking. Although the models may differ in approach, they tend to be based on the same concept: Creative thinking is a process which alternates between convergent and divergent methods of reasoning.

The Universal Traveler (Koberg and Bagnall, 1976) presents an example of a good, conventional creative problem solving model. The model is comprised of seven stages: 1) Accept, 2) Analyse, 3) Define, 4) Ideate, 5) Select 6) Implement and 7) Evaluate. The problem solver interacts with the model as if it were a journey designed to lead him or her through the process. Stages are comprised of exercises and games. They can be accessed in any order, depending on the needs of the specific problem being solved.


Electronic Thinking Cap
COMPUTER ENHANCED CREATIVE PROBLEM SOLVING
I've got an Idea!
IMPLEMENT
TIME/TASK CALENDAR
IMPLEMENTATION CHECKLIST
NAME IT
Hard Copy
Certificate
SELECT
WEIGHTED MATRIX
criteria for successful
solution verses
idea options
VISUAL SIMULATIONS
visualizations using
graphics peripherals
IDEATE
MANIPULATIVE VERBS
visualizations using
graphics peripherals
VIEWPOINTS
VISUALIZER
visualizations using
graphics peripherals
VIEW
attribute lists and
forced relationships
DEFINE
KEY WORD HIGHLIGHTER
defining and redefining
the problem statement
FACT-FIND
VIEW
window into
data base
P.F.S. TM
data organization
LIBRARY ACCESS
dial-up access to
collections
ELECTRONIC NETWORKS
The Source, CompuServe,
Delphi, etc
MOTIVATE
WARM-UPS
games and brainteasers
WEIGHTED MATRIX
personal priorities
verses
problem demands
EVALUATE
PROGRESS DOCUMENTATION
history of interaction
EVALUATION TOOL BOX
quantitative
or
qualitative response
THE SEVEN PHASES OF
CREATIVE PROBLEM SOLVING
APPLICATION MODULES
start here
H.D.S. ATTRIBUTE HANDLER
•
Foundation Program
optional power source

Directions:

1. Cut along solid line
2. Insert tabs in slots
3. Attach optional power source
4. Insert Head

FIGURE 1. (Metros, 1983) ©1983 IEEE.

BEST COPY AVAILABLE

DISADVANTAGES OF CONVENTIONAL CREATIVE PROBLEM SOLVING

Creative problem solving is a difficult concept to teach. First, the logical progression through steps is oftentimes awkward for visually oriented, intuitive thinkers. Secondly, the process can be time consuming and the problem solver tends to "hurry it up." One of the major objectives of creative problem solving is to generate a quantity of options. "The more methods you have time to use, the better off you will be in the long run. If you could travel by foot, auto, bus, and plane, the combined view would be broader than by any one means (Koberg and Bagnall, 1976)." A user that is in a hurry, tends to generate fewer options, and, thus, defeats the purpose of the creative problem solving process. Lastly, a conventioanl creative problem solving model is set up, on the most part, for non-visual interaction. A designer needs to be able to think visually in order to fully conceptualize ideas.

THE SOLUTION

The purpose of my research is to develop a creative problem solving model that addresses the needs of a visually oriented user. The microcomputer has tremendous potential as a tool for both improving and enhancing this type of user's creative problem solving skills. The reasons are as follows: 1) Information can be logically and sequentially processed. 2) A quantity of options can be generated in a comparatively short period of time. 3) One-to-one interaction is always available. 4) Information pertaining to the problem or solution can be easily stored, retrieved, organized, and manipulated. 5) Idea options can be evaluated more objectively. 6) A system can be augmented with graphics software and/or peripherals so that idea options can be visually manipulated. This provides the user the opportunity to set up "what if?" simulations.

My research is focused on microcomputer, not mainframe, applications. The model runs on the Apple II series of computers. Many professional designers and students have access to personal computers in their work places, classrooms, and homes. If the computer is to be a cost effective and a labor efficient investment, it should address both the practical and the creative needs of the user. The microcomputer used in this research enhances the PROCESS of design. It is not being used to produce a finished piece, such as computer art. Therefore, the more costly, sophisticated systems; that produce high resolution imagery and output; are not required for this application.

THE RESEARCH PROJECT

Electronic Thinking Cap is the title of my model for microcomputer enhanced creative problem solving (Figure 1) (Metros, 1983). It was funded under a research grant awarded to me by Michigan State University. Similar to traditional creative problem solving models, Electronic Thinking Cap employs gaming techniques to playfully coax the user into manipulating information in ways that new combinations and fresh ideas surface. The system consists of a foundation program and seven application modules. The foundation program (H.D.S. Attribute Handler) is an open-ended database that permits the user to add, search, retrieve, modify, and string together attributes. The application modules correspond to the seven stages of a conventional creative problem solving model. Each module is comprised of games, services, and methods that permit the user to fully explore the problem and experiment with solutions. In some cases, commercially existing software has been integrated into the model and, in other cases, software has been authored by computer programmers to fulfill specific needs.

The following is a brief description of the seven application modules (Metros, 1985):

1. MOTIVATE: Perceive the problem as a worthy challenge and accept the responsibility to try to solve it.

2. FACT-FIND: Collect any information and/or previous knowledge that could pertain to the problem.

3. DEFINE: Interpret and clearly state the problem. Sometimes the problem needs only to be redefined and restated in

order to be successfully solved. Designer and educator, Bob Gill stresses, "The most likely way to find a unique solution is to find something unique about the problem...The problem is the problem (Gill, 1981)."

4. IDEATE: Generate a QUANTITY of solutions to the problem, regardless of quality and practicality.

5. SELECT: Formulate a list of criteria for successful solution. Compare and choose the idea option(s) that best fit the criteria.

6. IMPLEMENT: Produce, or arrange for the production, of the selected idea so that the results can be disseminated to the targeted audience.

7. EVALUATE: Quantitatively and/or qualitatively measure the success or failure of the idea produced.

CONCLUSION

Electronic Thinking Cap is still in the developmental and testing stages. Right now it is being used, on an experimental basis, by myself and some of my graphic design students. I am also in the process of adapting it to an IBM system with a broader range of graphic capabilities. I hope to eventually make it available to interested educators and practitioners. The model is not limited to artistic applications. It is generically designed and can be used in any profession or discipline that practices creative decision making.

Artists have voiced concern that computer enhancement of creative problem solving might eventually impede their ability to think creatively. However, the computer is not a substitute for innovative thinking, nor can it generate its own ideas. Massimo Vignelli, a designer of international renown, eloquently and simply summed it up, "The computer frees the designer to design (Vignelli, 1982)." Electronic Thinking Cap will not replace conventional creative problem solving, but it can enhance the process by providing an interactive, fast-paced and challenging experience.

REFERENCES

Fahnstrom, D. "Technology, A Necessity in Design Education," The Design Journal, 1983, Vol. 1 (2), p. 11.

Gill, B. Forget all the Rules You Ever Learned About in Graphic Design. Including the Ones in this Book (New York: Watson-Guptill, 1981) p. 11.

Koberg, D. and Bagnall, J. The Universal Traveler (Los Altos, CA: Kaufmann, 1976).

Metros, S. E. "Electronic Thinking Cap: Microcomputer Enhanced Creative Problem Solving," Leonardo, 1985, Vol. 18 (2).

Metros, S. E. "Microcomputer Assisted Creative Problem Solving," Proceedings: Third Symposium on Small Computers in the Arts (Silver Springs, MD: IEEE Computer Society Press, 1983) pp. 30-31.

Naisbitt, J. Megatrends (New York: Warner Books, 1984).

Osborn, A. Applied Imagination (New York: Scribners, 1963).

Vignelli, M. Proceedings from AGI Viewpoints Conference (Toronto: Alliance Graphique Internationale, 1982).

Melodic Manipulations: Teaching Composition With Computers

Rene Upitis
Queen's University
Kingston, Ontario
Canada K7L 3N6

Two music software programs, Musicland and Melodic Manipulations, are explored as tools for teaching the elements of music composition to Grade 3 students. Specific teaching strategies and findings are described. Findings indicate that children are able to learn music notation and such concepts as timbre, transposition, and retrograde, by interacting with the computer software in the context of a structured but flexible teaching setting. Most importantly, the students have found music itself to be an interesting subject of study.

Few would dispute that most elementary school music programs fail to provide a challenging musical environment for students. Many children leave public school with only a repertoire of rote songs and a smattering of music theory. This training does not give students enough knowledge and motivation to continue musical pursuits in their adolescent and adult life. Indeed, often it is only the students who have had the benefit of private instrumental lessons who are likely to continue with music at the high school level and beyond. However, the music discipline as a whole should not and need not be limited only to those children fortunate enough to have the benefit of private music instruction. By using the powerful capabilities of the microcomputer along with software which requires children to think about musical concepts, many more children can be given a long-lasting music education.

The research described here is unique in that music composition by computer is used as the primary vehicle for achieving music understanding. Music composition as a field of study has been neglected or ignored not only in public schools but also in private study. There are two strong reasons for this neglect. For one, composition has been regarded as an activity only for very talented or highly trained individuals, usually adults. This attitude, in fact, may be caused in part by the second reason: traditionally, the skills of composition take years to develop. Yet, composing music need not be reserved for only a few individuals because of these reasons. After all, teachers expect and encourage original work in other subject areas. Indeed, one would think it ludicrous for students to learn English composition simply by reading the works of others. How can children be expected to understand the music discipline if the best that is offered is mastery of others' compositions?

The difficulty in teaching music composition is not so much a conceptual problem as a practical one. Even if teachers feel that it would be worthwhile to teach composition, how can they make it feasible to develop students' abilities in music composition and music as a whole?

There are several characteristics of the microcomputer which make it possible to create a musical environment where children are able to explore the elements of musical composition. First, the computer makes it possible to make accurate records of musical ideas. Second, the entry of musical ideas can be achieved without years of prior musical training, particularly with systems which use gestural input devices such as the Koala Pad. Third, computer software makes it easy to manipulate musical ideas. In this kind of environment, children can learn to test hypotheses, predict results, make musical transformations, and learn how the various elements such as rhythm, pitch, line and texture act on each other in forming musical structure.

Method

The research described here shows how specific teaching strategies can be used with music software for the Apple microcomputer to give children the skills and motivation to explore musical ideas through composition. Two music software packages, developed by Dr. Lamb and his colleagues at the Computer Systems Research Institute, University of Toronto, are used as tools for teaching music composition to the children of two Grade 3 classes in a public school in Kingston, Ontario. One classroom uses the Melodic Manipulations program exclusively, while the other classroom uses both Melodic Manipulations and Musicland. Both software programs give children tools to create and edit music compositions by manipulating the "raw" materials of pitch, duration and timbre. They differ in the amount of music theory knowledge needed to use the programs. Musicland can be used almost immediately without prior music instruction, while Melodic Manipulations requires some understanding of standard music notation.

A pre-test indicated that the children had few musical skills. Only 13% of the children, all of whom were receiving private music instruction, scored over 50% on a simple test containing items on music notation (e.g., duration of quarter, eighth and half notes and pitches in the treble and bass clefs) and concepts (e.g., melody and timbre). However, an ear test indicated that most students were able to accurately identify the major or minor quality of musical excerpts. Also, while a large number of the students thought that they would be unable to compose, they seemed willing to give composition a try. Finally, all of the children indicated that they enjoyed music in one form or another, and every child was readily able to name an instrument that he or she would like to play. So, clearly the possibility of teaching these children more about music in general through the specific medium of composition was - and remains - a strong one. Like most Grade 3 students, they had little knowledge at the outset, but their interest and natural abilities indicated that they would participate successfully in the music composition program.

The children were taught in small groups (up to 7 in a group), for one-half hour, every second day. They also worked unsupervised, in pairs, with the music software for up to one hour a week. During the instruction sessions, the children were involved in activities which help them learn music notation and concepts in the context of composing music using the software programs. These concepts and notation were taught using a variety of methods, both with and without the computer. For instance, timbre (the special quality of sound associated with each instrument) is a concept which was taught using Musicland, specifically through the Timbre Painting and Sound Factory games. However, rhythm notation and patterning were taught without the computer, using gross motor activities and spatial analogs for time. One of the most successful motor activities was the "body canon". Children were asked to repeat a four beat patterns "played" on some part of the body (e.g., quarter, quarter, eighth, eighth, quarter played on the

BEST COPY AVAILABLE

knees) by the leader. Meanwhile, the leader plays the next pattern, and so on. Thus the leader begins and ends four beats before the rest of the group. The body canon activity was also used to introduce other time signatures, such as 2:4 and 3:4 time. The work with body canons not only gave children a physical expression of rhythm, but also made it much easier for the children to listen for and write canons using Melodic Manipulations. Another important aspect of notation taught to the children was pitch notation in the treble clef, along with the corresponding notes on the keyboard. This enabled the students to compose music with Melodic Manipulations, and also to compose music on staff paper and play it on the piano when the computer was not available. Instruction also included differentiating between major and minor (using the piano), melody and chord (using Musicland), and learning about transformations such as transposition, diminuition, retrograde and augmentation (using Melodic Manipulations). Frequently the instruction sessions ended with 3 or 4 minutes of sharing of compositions or singing around the piano.

Most of the time spent on the computer without a teacher present was unstructured. In other words, children were encouraged to use the software in any way they chose. However, if students found the "create whatever you like" instruction too open-ended, they were given ideas or challenges. For example, for the Musicland program, they were encouraged to make a crashing sound, or a sound like gentle rain.

Findings

As expected, Musicland was easier for the children to use and understand than Melodic Manipulations. It appears to be a valuable tool for teaching the eight- and nine-year-olds about melody, chord, texture and timbre. The most enjoyable and instructive games appear to be Music Doodles and Timbre Painting. However, without additional instruction regarding more conventional sounds, interest in Musicland diminishes.

Although it was expected that Melodic Manipulations would be more difficult for the children to use than Musicland because of the use of standard notation in Melodic Manipulations, it was found that there was even more difficulty than expected. Many of the children found it difficult to control the Koala pad, especially for single note input. Children were also often confused by the text, since many of the terms are beyond their reading level. This difficulty lessened after about four weeks, partly because the children were asked to enter their own music in group sessions, and partly because of their increasing familiarity with the program. Some changes in the text for Melodic Manipulations would be appropriate for children of this age group.

The students learned the fundamental aspects of notation, namely simple rhythm notation and pitch in the treble clef, in less than six weeks. However, it should be noted that most of the learning did not occur at the computer. While the computer was a useful tool for introducing relative concepts (e.g., fast vs. slow or high vs. low on the staff), the specific durations (e.g., quarter vs. eighth) or specific pitches (A vs. F) were learned without the computer. Most of the rhythm relationships were best taught using gross motor activities and a spatial analog for duration. The pitch notation was taught primarily through repetition, using both the keyboard and the computer to reinforce the pitch notation.

A process which appears to take several months to develop is that of refining a composition. Most students payed less attention to the form of their melodies than was originally anticipated. A number of students filled many pages with technically correct notation, but were unable to predict what the melody would sound like. In fact, these children did not plan their melodies, but merely filled manuscript paper with notation. Even though these children

BEST COPY AVAILABLE

appeared to have no idea what they were writing, they were encouraged to carry on in this fashion for several weeks. The reasons were several. Perhaps the most important reason was that they were thoroughly enjoying this activity, spending much of their spare time in and out of school "composing music" and were beginning to view themselves as composers. Second, writing reams of notes gave these children considerable practice in notation, in a sense, providing them with an "immersion" setting for music. Third, the notation gave the teacher something concrete to work with when in trying to teach structure or form.

One of the first examples of refining a composition was given by a child who was writing a piece in D minor. The first version ended on F. Commonly, music ends on the tonic or key note, which in this case would be D. The child played his piece on the piano, and afterwards erased the F, changing it to a D. This demonstrates that some of the "rules" of melody writing can be discovered by the children through the process of writing music themselves. The process of editing can be encouraged by providing a setting for performing and revising one's compositions. In such a setting, children can be given an opportunity to play their pieces, and to explain the mood or intent of the composition. Other students comment on the work, offering suggestions as to how the work might be improved.

Many of the children wrote about their music composition activities in their daily journals. This is a significant finding. It indicates that the music activities stretch beyond the classroom, and have found a place in the children's repertoire of worthwhile things to think and write about. Another indication that the music teaching is having an effect on the children beyond the immediate teaching setting is given by an example of one child's art work. The children were shown how to make angels as Christmas decorations. Several students, unprompted, made music for the angels to hold, drawing a staff and placing notes appropriately on the staff.

Some of the compositions created by the children on their own were accompanied by words. None of the children were specifically asked to write words, but on their own, began writing songs. This indicates that they are relating their new music teaching to their repertoire of musical experiences, presumably enriching both their present and former activities in the process.

Children's interest in other instruments has also been enhanced. They were keen to hear their compositions on "something different", that is, on something other than the computer or the piano. Many have expressed the desire to learn to play an instrument themselves.

As a final comment, one of the most promising findings to date is that the students have become interested in music as a subject of study. While the computer offered a great deal of motivation to the students initially, the sustained interest appears to come from the **music** itself. The computer quickly became just another tool in the music classroom. Students were just as eager to use the piano or tap rhythm patterns on coloured paper as they were to use the computer. Thus, the computer may indeed be used as an effective tool for teaching the elements of music and music composition, but only if the students are guided through the creative process with structured and flexible teaching.

This Research was funded under contract by the Ministry of Education, Ontario. This report reflects the views of the author and not necessarily those of the Ministry.

BEST COPY AVAILABLE

It May Be Courseware, But Is It Educational?

Brian Durell
University of Toronto
371 Bloor St. W.
Toronto, Ontario M5S 2R7

Teachers seldom plan lessons in terms of a theory of learning. None the less, all teaching reveals an underlying theory of learning. The same is true for educational computer programs. In this paper we examine common types of educational computer programs in terms of theories of learning. On that basis, problems in present educational computing are identified and promising directions for future work are suggested. One conclusion is that the apparent scarcity of quality educational programs may be due to looking in the wrong direction.

An analysis of a teacher's activities in the classroom can reveal the theory of learning that the teacher believes in. It is not necessary for the teacher to be aware of an explicit theory. In fact, a teacher may give mental assent to one theory yet teach according to another. Educational computer programs also reveal underlying theories of learning.

Most of the programs which teachers use on classroom computers are conveniently classified as computer-assisted-instruction (CAI). The archetype of CAI programs is the tutorial style which is intended to mimic traditional teacher actions. Tutorial programs do not assume that a student has prior experience with the subject matter of the lesson. Therefore there is a need for the program to provide some information. Next, a tutorial program provides questions concerning the information. The answers to the questions are judged by the program and appropriate feedback is given to the student.

Writing tutorial programs is a lot of work. They require thorough analysis of a subject matter and an understanding of the difficulties which students commonly experience. It is one thing to deal with student difficulties in face to face teaching and quite another to anticipate such difficulties with no students present. Also, once difficulties are identified, a tutorial program is expected to do something about them. That is, a tutorial program must provide remedial material. It is not surprising then to find that most of the educational programs available for school computers are NOT of the tutorial type.

It is much easier to write drill and practice programs. Programs of this type assume that the basic teaching has been done elsewhere - perhaps by another computer program, by a teacher in a traditional classroom, by a programmed instruction book. A drill program provides questions for students to answer based on previously taught material. Although good drill programs will deal in some manner with student difficulties, most do little more than provide right/wrong feedback. This greatly simplifies the programmer's task in preparing the program. The initial information provision is assumed to be done elsewhere. The requirement to provide remedial segments is removed. Therefore, drill programs deal only with the question-asking and answer-judging components of a tutorial program. There is little to distinguish most drill programs from simple tests.

It is little wonder that most educational computer programs are of the drill type.

Educational computer games are also popular applications. Many games are more or less disguised versions of drill programs. Some games can get students involved with skill drills in a manner which students find much more acceptable than an explicit drill program. Computer games are not limited to this approach, as we shall see later in this paper, but the majority of the computer games in use in classrooms now are either dressed up versions of drill programs or of little educational value.

Tutorial and drill programs are in line with a behavioural theory of learning. Modern behavioural theory, based on operant conditioning, sees learning as something which happens to a learner. That is, influences in our environment have effects upon us which make us more likely to do some things in the future and less likely to do others. To a behaviourist, learning consists of changes in behaviour. If a person becomes more or less likely to perform a particular action, the person's behaviour has changed so learning has taken place. In a behavioural approach, learning is heavily dependent on the provision of stimuli to a learner and then provision of feedback, usually interpreted as positive reinforcement, after a learner does something. A computer program which provides a stimulus on the screen, waits for a response at the keyboard and then generates feedback for the user clearly fits the behavioural model.

Behavioural theory can explain a surprising amount of human behaviour with a very small number of assumptions. There is a strong attraction to a theory of that sort. However, the emphasis of behaviourists on the objective and observable leaves internal human processing of information and personal motivation largely outside the realm of study. This is uncomfortable for most teachers who recognize that students are not passive creatures waiting for teacher stimulation and highly dependent on teacher feedback. Human beings are very actively engaged in their learning. They seek some information and ignore other information. Different individuals experience essentially the same situation and make entirely different things out of it. Therefore, useful though it is for some learning, behavioural theory does not cover a broad enough range to serve as the sole theory of learning to explain all that teachers wish to accomplish.

Cognitive approaches in psychology deal with the internal processing of information which is an important part of learning. Cognitive theories recognize that different people will react to indentical stimuli in different ways. Teaching based on cognitive theory is not concerned so much with presenting the same stimulus to every learner and trying to get the same response from all learners. Cognitive approaches treat learners as individuals who are highly involved in constructing their own understanding of the world. Education, in these terms, is not something which is done to you, but something which you do for yourself.

Simulations are educational computer applications which have some of the characteristics of behaviourally based learning and some of cognitive approaches. Simulations may replicate physical science experiments or social science experiences. Usually teachers intend for students to learn the model underlying the simulation. In that sense, a simulation has a message and is intended to communicate some definite content to students. In this way it promotes behavioural theory based learning. At the same time, a simulation gives students opportunities to explore within the context of the simulation model and, to some extent, gain more control of their learning than is usually true in a typical didactic lesson. The discovery aspect of a simulation is in line with cognitive theory approaches to learning.

Piestrup (1984) has suggested ways in which computer games can provide more than palatable ways of drilling students. In her plan, games should be constructed in series, not as standalone items. Concepts used in one game should transfer to others so learners can develop a general set of skills which are useful in a range of games. Such generality is an important component of a set of games. Although students will learn some content from a set of games, the more important outcome is learning strategies for further learning. Games developed in this manner are compatible with cognitive theory interpretations of learning.

Papert (1980) has presented LOGO as a computer language which puts computing power into a student's hands to be used for reaching the student's own goals. Papert refers to this as letting the student program the computer instead of using the computer to program the student. This approach is clearly aligned with an approach to learning based on cognitive theory.

Programming in LOGO is not the only way for students to use computers to further their own learning. Word processing is a widely acclaimed aid for students. A word processor does not have an explicit lesson to transmit to students. It provides power for students to express themselves and to easily revise their expressions. It encourages clear thinking and mastery of written communication. Use of a word processor can easily go beyond merely doing the old things in a new way. Students can write knowing that they can revise their work easily. Revision becomes an accepted and expected part of expression rather than a lofty but seldom realized ideal.

Spreadsheet programs also have potential for educational applications. Since they are designed for use by business people, they have obvious potential in business education. A student can construct many balance sheets using a spreadsheet program and get immediate feedback on whether or not the figures are in balance. Similar use may be made of spreadsheets in learning basic concepts in inferential statistics. Models of scientific phenomena can be expressed in spreadsheet formulae and an easily-constructed simulation may be executed to provide a dynamic demonstration of what the scientific model claims.

Both word processing and spreadsheets were designed with business applications in mind, not educational applications. That they have educational significance is a happy accident. The main value of word processing and spreadsheet programs may be to serve as models for educational software which is similarly designed to allow students to manage their own learning in a more efficient manner. What is needed is educational equivalents of VisiCalc and WORDSTAR programs.

Before the days of microcomputers, Ellis (1974) observed that "The more crucial aspect of the computer's promise for education rests in the opportunity - and the responsibility - it provides us to analyze and to better understand the very processes we automate" (p. 42). A computer can be used to automate any process which we care to or are able to describe to it. That is, we can carry on business as usual with a computer simply doing things more quickly and perhaps more efficiently. The real significance, however, of computers for education lies in the possibility of doing things with them which we could not do without them. Ellis has further stated, "**... thinking about the computer's role in education does not mean thinking about computers; it means thinking about education**" (p.42).

If we seize the opportunity which Ellis has suggested is before us, we will be less inclined to accept programs which simply cast computers in the role of purveyors of information. We will be less inclined to ask for more finely tuned tutorial and drill programs. We will, instead, look for programs which stimulate students to think and which students can use as aids in building their own understanding of the world. In this direction lies the real potential for computers to make a significant impact upon education and for education to gain a fresh forward thrust. Computers may serve mainly as catalysts to help us take a hard look at what we are doing in schools and to capitalize on our opportunity to change traditional practices in ways which reflect our understanding of how people learn.

REFERENCES

Ellis, A. B. The use and misuse of computers in education. New York: McGraw-Hill, 1974.

Papert, S. Mindstorms. New York: Basic Books, 1980.

Piestrup, A. Game sets and builders. Byte, 1984, 9, 6, 215-219.

Custom-Designed Lessonware For Second Language Learning

D.M Paramskas
University of Guelph
Department of Languages
Guelph, Ontario N1G 2W1

Although excellent software is now available for second language teaching, there are some drawbacks. In the case of pre-programmed courseware, a specific methodology is imposed and the programs cannot be modified; in the case of authoring systems, the complexity still discourages the average teacher. One possible solution is the template: simple in usage, requiring no computing expertise, yet producing highly sophisticated lessonware. Using four different kinds of templates, we will demonstrate how to exploit the template format in order to allow teachers to easily create CAI materials suited to various learning approaches and levels of skill.

When computer-assisted instruction first appeared in the early 60's, the hardware available was complex, cumbersome and to many, awesome. Progress in technology, however, was swift, and in the last few years, exceedingly so. Hardware today is relatively simple, user-friendly, and attractive even to teachers determined to resist the juggernaut of technology in order to preserve the humanistic aspects of pedagogy.

Software has evolved in a similar fashion. The first programming languages, such as COBOL or FORTRAN, were tools much too complex for the average teacher to learn and use. Authoring languages such as BASIC were designed to be of much easier access to the non-specialist, but were still enormously complex for the design of pedagogically acceptable programs. Authoring systems appeared shortly after the birth of the microcomputer, and simplified programming to the point where no specialised training was required. Teachers began to reconquer the field of software and produce programs which met their exacting standards.

Several problems remained, particularly in the domain of second language learning. Most authoring systems were designed to create courseware which focussed on the learning of facts or the solution of problems, particularly in the sciences and social sciences. The question of form learning, the elicitation of input from students which is to be judged not only for substance, but for the form in which substance presented itself, was not really addressed by systems such as DASHER, SuperPilot and MICROSCOPE. These could indeed be used for form learning, but only awkwardly so, and with severe limitations as to format and error checking. In the large majority of cases, the format is limited to single sentence cues, single responses, and error checking consists of a list of variable length of anticipated errors.

A second problem is less technical than psychological. Authoring systems do need some effort on the part of the teacher for complete mastery, much like word processing software, or, for that matter, learning a second language, and mastery must be constantly practiced or it will rust away. Teachers in this day and age simply do not have the time to dedicate to this type of on-going practice, unless they transform themselves into authoring specialists, to whom other teachers turn when they want their designs programmed. Teachers would then again be divided into two groups, the specialists and the front-line troops, with the usual problems in terms of communication, pedagogical approaches and styles which occur in such a situation.

This is not in any way meant to criticize courseware created by specialists, some of which is superb. CAI courseware can be acquired much as any other commercial learning support materials are: textbooks, audio-visual materials, etc. This requires a considerable amount of shopping around, which is not easy when sample programs are still rarely available for previewing. Some excellent commercial courseware exists for second languages: APFELDEUTSCH for German, CLEF (Computer-assisted Learning Exercises for French) for French, to mention only two large courseware packages. It is, unfortunately, impossible for a non-programmer to modify these in any useful way in terms of customizing.

Progress marches on, however, and a modified form of authoring systems has been developped, called the template. A template is a mini-authoring system designed specifically for one learning area, and within that area, for one format at a time. It is completely user friendly, uses natural language for prompts, and does not, like regular authoring systems, require the learning of abstract commands or symbols. Its narrow focus allows for a great depth in complex error checking, some of which is inbuilt and automatic, some of which can be custom designed by the teacher. The formats can range from whole stories, through paragraphs, single sentences, or visual cues. A teacher would merely choose the format preferred for a particular learning task, and be able to program a lesson in the time it takes to type in the data. It should be emphasized, however, that preparation time for the lesson is in no way reduced: as for any other medium, the pedagogical soundness of an exercise is proportional to the time invested in its preparation. Templates are merely tools, perhaps analogical to a super-typewriter, which permit the exercise to function on a computer.

Most templates now appearing are of the type called "open-ended courseware" (Canale et al. 1984)that is, a combination of a fixed program written by a specialist, plus a template which allows the teacher to modify that program or create a variant on it. Two good examples of this type of courseware for ESL are CLOZE and Storyboard . Four other types of templates will be examined in this presentation. Two are designed for reading comprehension, but may be broadened for other uses; they function for English, French, Italian, Spanish , German and Latin as second languages (PROMPT: Program for Reading cOMPrehension : Teacher-controlled). The other two are for dictation (DICTATE) and for oral comprehension, verified in writing (LISTEN); they also can be used for all the above languages.

Templates are not meant to replace specialized courseware, but to broaden the range of tools available to teachers. They permit the customization of exercises to fit the pedagogical philosophy and methodology of the individual teacher-user. In the more rigid, non-modifiable courseware, an approach is usually imposed upon the teacher, be it the structuralist one, or variations on the communicative ones. It seems to us that good teachers should be able to create CAI materials suited to their own approach, whatever that approach may be, fashionable or not as it may be. Good teaching is not approach dependent; it is not assured because one follows the dictates of the latest research findings into language learning. Good teaching is very much an art, one which depends on the teacher's having access to educational tools which further that art, tools which suit his or her approach and philosophy.

While the future holds the promise of varied marvels such as interactive video and sophisticated parsing systems, these will not be available on a broad basis for some time. Templates are available now, and are being developed in increasing numbers. Teachers who wish to create materials for immediate use need only learn to design well, using their pedagogical expertise, not knowledge of programming. Well-designed lessonware can be accumulated, and need not become obsolete when more sophisticated systems appear.

REFERENCES

Canale, Michael, G. Barker, M. Bélanger, K. MacRury, R.S. McLean and R.G. Ragsdale. 1984. _Microcomputer Software for the Language Arts in Ontario: Survey and Analysis._ Toronto: Ontario Institute for Studies in Education.

CLEF is a joint project of the universities of Western Ontario, Guelph and Calgary, and runs either on Commodore or IBM. For the Commodore version, contact Glyn Holmes, Language Laboratories, University of Western Ontario, London, Ontario. For the IBM version, contact D. Paramskas, Department of Languages, University of Guelph, Guelph, Ontario.

CLOZE and Storyboard are designed by Christopher Jones and available from Holt, Rhinehart and Winston.

PROMPT, LISTEN and DICTATE are joint projects of the universities of Calgary and Guelph. For information, contact D. Mydlarski, Department of French, Italian and Spanish, University of Calgary, Calgary, Alberta.

Micrometrics: A Metric System Tutorial

Anthony P. Malone
Raymond Walters College
University of Cincinnati
Cincinnati, Ohio 45236

Micrometrics is a computer based instructional program that is especially designed to give the student a clear understanding of the major units of the System International (SI). The presentation is aimed at the upper elementary grades and prospective elementary school teachers at the college level. The program has been successfully implemented on several microcomputers.

The idea for a computer based tutorial on the Metric System first came while I was teaching a Math for Elementary Teachers course. A unit on the International System of Units was a major part of the curriculum. The students in the course had some problems with the metric base units and prefixes, so I decided to write a short program to teach this material.

The initial program was designed to run on the TRS-80 model I microcomputer with 4K memory in 1978. The first module covered the metric system base units, metric prefixes, and simple conversion in the SI units. The program was tested in the classroom using prospective elementary school teachers.

In 1980 the program was fully revised and expanded to 32K for the TRS 80 model I. This second phase of development was the result of a project for a graduate course for teachers entitled Computers in the Instructional Process.

The second phase of program development involved the use of simple computer graphics to teach the metric system. At this stage the concept of the "metric ladder" was first used to teach the metric prefixes and conversion. These preliminary metric system tutorials used the BASIC progamming language without color graphics.

In 1983 I decided to rewrite the tutorial for the Metric System in the Pascal language. At this point the program had to be completely modularized and redesigned. The VAX 11/780 microcomputer was utilized for this purpose. A series of ten different units on the metric system were developed to run successfully on the Vax in Pascal.

The next step was to recode program to run on the Apple IIe in Apple Pascal. The ability to write procedures in Apple Pascal was hampered by the unwieldy Apple Pascal editor and Compiler. I then knew that BASIC was the best language for the tutorial. Apple Pascal featured the Applestuff and Turtlegraphics packages. These routines allowed me to add simple graphics and sound to further enhance the presentation of the metric system.

The present version of Micrometrics runs on the Commodore 64 using the Graphics Basic package designed by Ron Gilbert and Tom McFarlane of HesWare, Inc. This graphics package is really a major enhancement of the Commodore 64 Basic. It is essentially a graphics tool kit that allows the programmer the freedom to design complex graphic routines with ease.

The most important feature of the Graphics Basic language for the construction of the tutorial was the Procedure statement. All routines were written as Procedures and called using the Do statement. Thus many GOSUBs, GOTOs, POKEs and PEEKs were avoided. The final results for the tutorial were outstanding. The BASIC editor on the Commodore 64 was extremely helpful in developing the various procedures. In addition, the total cost of the Graphics BASIC package is under $25.

All the major strengths of the Commodore 64 were used in the final Micrometrics tutorial revision. Color graphics, sound and sprite animation were easily added to the program to make the presentation enjoyable and interesting for the student.

The following units were developed in 1984 using the Hesware Graphics Basic diskette for the Commodore 64:

Micrometric Unit Zero
A unit containing all common procedures and variables.
(filename metriczer)

Micrometric Unit One
A unit that presents an introduction to Micrometrics.
(filename metricone)

Micrometric Unit Two
A unit that presents the major base units of metric system.
(filename metrictwo)

Micrometric Unit Three
A unit that presents a short quiz over all of the SI units.
(filename metricthr)

Micrometric Unit Four
A unit which presents the metric prefixes and abbreviations used in the SI.
(filename metricfor)

Micrometrics Unit Five
A short quiz over the metric system prefixes and abbreviations.
(filename metricfiv)

Micrometric Unit Six
Drill and practice on elementary conversions in the metric system.
(filename metricsix)

Micrometric Unit Seven
A unit on temperature and time measurements in SI.
(filename metricsev)

Micrometric Unit Eight
This unit presents drill and practice on advanced metric conversions.
(filename metriceig)

Micrometric Unit Nine
This unit presents measurements of area and volume in the metric system.
(filename metricnin)

Micrometric Unit Ten
This unit simulates metric races at the Olympics. A 1000 metre run for Meterman and Literman is presented. A 100 metre run for Gramgirl and Tempgirl is presented. Sprite Animation is employed.
(filename metricten)

The major instructional objectives of the metric system tutorial are:

1. The student will recite the names and abbreviations of the base units of the SI.
2. The student will write the names and abbreviations of the Greek and Roman prefixes.
3. The student will perform simple conversion in the metric system for various base units.
4. The student will perform advanced conversions by steps of 1000 for various base units.
5. The student will distinguish measures of area and volume in the metric system.
6. The student will compare and contrast temperature scales and measurements of time.

References

Conversaphone Educational Series. The Metric System Made Easy

Gilbert, Ron and McFarlane, Tom. Graphics Basic User's Manual. Hesware Brisbane, CA

Graham, Malcolm. Modern Elementary Mathematics. Harcourt pp. 295-303, 1975

Hume, J.N.P. and Holt, R.C. USCD Pascal: A Beginner's Guide. Reston Publishing 1982

Leuhrman, Arthur and Peckham, Herbert. Apple Pascal: A Hands-On Approach. McGraw Hill 1981

Meserve, Bruce E. and Sobel, Max. Contemporary Mathematics. 2nd ed. Prentice Hall. pp 302-316, 1977

Tucker, Allen B. Apple Pascal: A Programming Guide. Holt, New York, 1982

Wylie, Joanne. Beginning Metrics. A Golden Readiness Workbook.

A Model For Implementing Computers (K-12) To Promote Learning And Literacy

Robert Rivers and Edward Vockell
Purdue University Claumet
2233-171st St.
Hammond, IN 46323

A five-step model for promoting the use of computers in both instruction and literacy is proposed. The model is currently being used by a major school district to promote the orderly integration of computers into its curriculum.

This model promotes the use of computers in the K-12 curriculum to achieve two distinct but overlapping goals: (1) the more successful mastery of the major goals of the overall educational curriculum, and (2) the development of a high degree of computer literacy among students completing this curriculum. Two elements are essential in the model: (1) curriculum integration of computer-related instructional activities, and (2) teacher training. The elements are interactive.

To promote both instruction and literacy, a five-step model is proposed:

1. A K-12 team of teachers must evaluate and become familiar with computer software that has potential for learning and literacy at their individual levels. This familiarity must encompass both obviously educational applications (CAI) and more general applications (databases and spreadsheets).

2. The teaching team must establish K-12 goals for computer literacy that can be accomplished, for the most part, in the regular classroom. All content areas in both elementary and secondary levels are potential areas for teaching computer literacy.

3. The teaching team must become proficient in a technique for integrating software for both content and literacy learning into their individual curricula.

4. The teaching team will then select software for both learning and literacy in their own curricula. The software should be correlated with the overall K-12 goals established in step 2. An attempt should be made to use as many software packages as possible for both literacy and learning.

5. The teaching team will develop an action plan to implement the goals of the curriculum. The plan should include a curriculum guide that contains: (1) instructional objectives and/or goals to be accomplished by software packages at each level of the school

curriculum; (2) computer literacy objectives to be accomplished by software packages to be used; and (3) a list of software packages that can be used for instruction, for literacy, or for both at each grade level.

This planning process is currently being used by a major northwestern Indiana school district to promote the orderly integration of computers into its curriculum. The authors are participating in this process and will present the results of this implementation at the meeting.

SPECIAL EDUCATION

Individual Educational Plans -
A Continuing Saga of Computer Support

Tommas J. Koehler David Barr
Patricia Edmister Arla Bowers
Montgomery County Public Schools
850 Hungerford Drive, Room 151
Rockville, MD 20850

This paper presents a description of the evolution of computer systems at Montgomery County Public Schools that support IEPs. The discussion covers educational and technological issues encountered since the project began in 1978. The evolution of the present system and plans for its future expansion are included. This paper is a follow-up to a preliminary paper printed in the AEDS monitor, April/May/June 1981.

With the passage of the Federal P.L.94-142, and the various state by-laws and regulations implementing the law, the Individual Educational Plan (IEP) has become a major facet in the life of almost every special education teacher and administrator.

The IEP process is a laborious one. Much thought and attention is required to create precise objectives and other IEP content, and the preparation and typing of the plan is very time consuming.

Formulation of these objectives is often beyond the time available to most teachers, and may call for skills which are not within their training. Also, consistency of objectives within a school system can never be achieved if educational objectives are formulated "ad hoc" by every teacher.

There is, therefore, a valid need to develop means not only to reduce work associated with IEPs but also to improve quality. These two needs supply the basic requirements for any computerized IEP support system. Based on these ideas, Montgomery County Public Schools (MCPS) began work on a computerized IEP support system in 1978. A preliminary analysis indicated that the development of a set of IEP instructional objectives was of first priority. In 1978, special education teacher specialists began the collection and refinement of varied educational objectives for the objectives bank, and a rudimentary computerized support system was developed to store, retrieve, and edit these objectives.

By the fall of 1979, several hundred basic instructional objectives had been collected. At that time it was decided to develop a prototype computerized IEP support system to demonstrate and refine the concept.

IEP SUPPORT SYSTEM PROTOTYPE (ORBIT-1)

An interdisciplinary project team was formed in the fall of 1979, to develop a set of requirements, assumptions, and constraints used to define the system. These were:

1. The computerized IEP support system would store and retrieve objectives arranged in a simple hierarchy. This hierarchy would appear as follows.
   Goal Area: To demonstrate an improved level of independence in activities of school life.
      Annual Goal: Will develop habits of cleanliness/ personal hygiene, and good grooming.

Short-term Goal: Will brush teeth.

2. Computer retrieval was planned originally to be by individual objective or by hierarchy -- i.e., selection of items "low" in the hierarchical tree would also retrieve the higher items.

3. The computer system would be "user driven". The special education staff would control both the data and the execution of reports via CRT terminals.

4. The system would be implemented in schools on a purely voluntary basis.

The technical approach used was determined by the then available MCPS software and the desire to develop the prototype as rapidly as possible. The ORBIT prototype (ORBIT I) was developed using IBM VS BASIC. This programming package ran on the IBM 370-158 mainframe computer and was accessible by IBM 3270-type CRT terminals. These CRTs were used for both data input and job execution. IBM utilities and simple COBOL programs were used for reports (IEPs).

The completed prototype was demonstrated in November of 1979 and became operational shortly thereafter. The operations of the system are summarized below.

1. Teachers participating in the project were provided with a printed "catalog" of objectives in the bank and IEP order forms. They referred to the "catalog" to select appropriate objectives, then indicated on the order form the selection of goal areas, annual goals, and short-term objectives, and prioritized them based on the needs of the student.

2. The completed IEP order form was keyed in via CRT terminals. The ORBIT system then performed the selection programmatically and formatted the printed IEP.

3. The printed IEP was then reduced and sent back to the requesting teacher.

FURTHER DEVELOPMENTS

In early spring of 1980, the ORBIT I system was demonstrated to the staff of the Maryland State Department of Education (MSDE). Based on the demonstration and subsequent discussion, the ORBIT concept was expanded to include linking objectives with instructional materials, competency-based tests, and instructional strategies. MCPS was awarded an MSDE grant to support further ORBIT development, including:

1. The number of objectives was expanded from 250 to more than 2000.

2. The development of objectives linked to materials, tests, and strategies for Reading was completed, and initiated in other areas, such as mathematics and social emotional skills.
3. Design of an expanded computer support system (ORBIT II) was completed, and a database structure created.

EXPANDED IEP SUPPORT SYSTEM (ORBIT II)

The improved computerized IEP support system (ORBIT II) was designed along the following lines:

1. ORBIT II was implemented using software which would allow integration of the application into the MCPS database, using Cullinet's IDMS system. The software chosen for the online data entry and inquiry functions was IBM's DISPLAY MANAGEMENT SYSTEM (DMS). COBOL was to be used to load data to the database and Cullinet's CULPRIT report writer for reports.

2. The ORBIT II system was designed to allow for online IEP selection and printing from remote school sites, as well as from the "ORBIT" office.

3. It was determined to print IEPs using "letter" quality printers.

The ORBIT II system was programmed and implemented in phases during the period from July 1980 to July 1982. The ORBIT II system was very successful, as evidenced by the more than 10,000 IEPs which were created during the period from September 1980 through July 1983.

STREAMLINED IEP SUPPORT SYSTEM (ORBIT III

Following the upgrade of the mainframe to an IBM 3033 during the summer of 1982, MCPS began using Cullinet's Application Development System/OnLine (ADS/OnLine). This product allows creation of on-line applications that can query and update the IDMS database. Efficiency and ease of future maintenance dictated that ADS/OnLine should replace the ORBIT II DMS System. The design for this system, which began in January 1983, called for a phased approach to programming and implementation. Phase I was to be a conversion of the ORBIT II SYSTEM to ADS/OnLine with several new features. Phase II was to provide storage of student IEP's on the database.

In October 1983, programming was completed for Phase I containing the following enhancements:

1. On-line editing, including database verification of the objectives entered for an IEP

2. New data fields were provided for IEP annual goals. These include the title of the person responsible, the evaluation criteria to be used, and the projected starting and completion dates. Previously this data had to be typed onto the printed IEP.

3. The display of the last goal area and annual goal entered, to aid the terminal operators in keeping their place when keying an IEP

4. The printing of objective numbers of the IEP, to make it easier to verify the objectives requested

5. An on-line transaction log of IEP's created

6. An on-line module which displays or prints, for a requested short-term objective, all test items in the database which (linked to the short-term objective) measure that objective, and instructional strategies and materials (linked to the short-term objective) can be used to teach that objective

7. An increase in the speed of the printer to 500 lines/minute

Phase I of ORBIT III was implemented in the Orbit office in November 1983, and in remote schools in February 1984. The response to Orbit III has been overwhelmingly positive. Users are able to key the selections for IEP's faster, and the on-line editing of entered objectives has eliminated the need to re-key incorrect IEP's. Over 6,000 IEP's have been created using the Orbit III System in its first seven months of operation.

The design for the second phase was started in January 1984. The primary purpose of Phase II is to allow student IEP's to be stored on the database. The design calls for this to be accomplished by creating a link between the current student record on the database and the various objectives ordered for that student on the Orbit Resource Bank of Objectives. Once a student's IEP has been created and stored on the database, it may be viewed, modified, or printed (the complete IEP, or selected goals and annual goals) as needed. The database link will be achieved by a record containing the school year that the IEP will be in effect. This structure will allow the development of an IEP for the next school year without affecting the student's current IEP. An additional feature will provide the ability to report student progress against the annual goals of a student IEP. The programming of Phase II is currently underway.

MSDE funding for Orbit III has resulted in the completion for four Orbit manuals:

1. ORBIT RESOURCE MANUAL which lists over 4,000 short-term objectives categorized by Annual Goal and Goal Area and explains how to include objectives on an IEP.

2. ORBIT TEST RESOURCE GUIDE which lists test items for 30 MCPS-approved assessment measures and shows cross-referenced short-term objectives.
3. ORBIT MINI-REPORTS which lists selected teaching strategies, instructional materials, and test items that are linked to individual short-term objectives in the areas of social/emotional and functional life skills.

CONCLUSION

The concept of using the computer to store produced IEP objectives for a specific student is well established. User and parent reaction to computer generated IEPs at MCPS has been excellent, and the system's use has steadily increased.

Computer-assisted IEPs (Part B) have been used in Montgomery County Public Schools since November 1979. Over 15,000 computer generated IEP's have been created using the Orbit systems. IEPs have been generated for students ages birth to 21 with the following handicaps: speech impaired, mentally retarded, orthopedically impaired, other health impaired, seriously emotionally disturbed, and learning disabled.

MCPS has shared the results of the ORBIT project with other school districts. All three ORBIT notebooks are available for purchase. Also available is a licensing agreement to use the content of the manuals. In addition MCPS is cooperating with other school systems in adapting the MCPS computer system for their own needs. An agreement with Hillsborough County Schools, Florida, for example, has resulted in the adaptation of the MCPS ORBIT II system to run in a DOS/VSAM configuration on smaller IBM machines. This system will be available in the near future for other school systems to acquire. For the near future, MCPS is designing a distributed system so that ORBIT's database of objectives can be downloaded to microcomputers for local use. As these new designs are implemented, MCPS will again be sharing more information about IEP's by computer support.

Computers and Special Students: Classroom Applications Beyond Drill and Practice

Francine Bederman
Peel Board of Education
Mississauga, Ontario

Special students, gifted as well as slow learners, often evince weak fine motor skills, minimal self-confidence, rudimentary self-esteem and/or an inability to attend to tasks. When the computer is utilized effectively as part of the classroom programme, it can assist in the remediation of these deficits while enriching the educational environment. As well, computer technology can provide the assistance necessary to enable many physically disabled students to be educated with their peers. This paper will explore practical ways in which the special education teacher may use the computer to promote learning.

Computers excite students! We are not certain why. This phenomenon may be related to the desire to control and the "high" that results when a difficult problem is mastered. Perhaps the challenge of improving one's skill, getting a "high score", or just the age-old desire to win, keeps young people glued to their monitors. Whatever the reason, we as special educators should take advantage of students' fascination with computers to remediate, to motivate, and to facilitate learning.

A. Computers and Learning

Drill and Practice

Much has been said and written about this often over-used application of the computer. For gifted students, drill and practice programs are usually inappropriate. There is little capacity for creativity or exploration. For slow learners or any student with a deficit in a specific area, such as mastery of addition facts, drill and practice programs offer a new way to provide the repetition necessary for learning. Students can work at their own pace in a non-threatening, entertaining, colourful environment. Teachers should endeavour to use drill and practice programs which - reinforce only correct responses, are free of negative comments about incorrect responses, provide the opportunity to repeat difficult areas, give users the option of seeking help within the program, allow students to proceed at their own speed, permit teacher input or modification to allow for individual pupil needs, and provide record-keeping functions to permit students and teacher to monitor progress.

Word Processing

One of the easiest and most creative ways a teacher can use the computer in the classroom is the application of word processing.Current word processing programs range from the extremely simple to the powerful and complex. However, a complex program may be simplified by teaching students only the functions needed to complete the task at hand, for example - how to correct errors, how to delete words and how to move lines. Word processing allows all students to produce a neat hard-copy every time. Special students (be they blind, slow, gifted or physically disabled) who have been limited and/or frustrated by their inability to write neatly, correctly, or quickly with pen or pencil, can use word processing to express themselves on paper. Corrections are readily accomplished.

Word processing is applicable across the curriculum. Stories, poems, articles, and class newspapers can be created on the computer.

Research projects, science experiments, current events, and historical biographies can be produced with the aid of a word processing program. Students and teacher may discuss a draft of the finished product and students might then be asked to revise their work. The need for revision is more willingly accepted by students when they are able to use a word processor to edit their writing. Pupils utilizing word processing are learning language by using language rather than by being taught language (Goldenberg, 1984). Increased expression of ideas in written form is frequently the outcome when word processing is made available to special students.

Problem Solving

Many computer programs involve students in problem solving. Such programs provide opportunities for decision-making and allow pupils to observe the results of their decisions. The excitement that comes with being in control, increased self-confidence and strengthened problem solving skills, often result from using this type of computer application. Adventure games such as Scholastic Microzine's Twist-A-Plot, Odyssey, and Troll's Tales are popular with students and encourage reading as well as thinking.

Papert (1980) posited that one of the positive effects of using the Logo language with students is that Logo places students in control. *They* are programming the computer by means of the turtle. Students who experiment with Logo experience true trial and error learning. Mistakes become learning opportunities. Logo users learn a great deal about moving an object through space, structured thinking, and breaking large problems into small, workable pieces (Wanderman,1984). Logo can be used with minimal direction or with specific instructions. It addresses the curriculum in the areas of shape, angle, directionality, art and language and its applicability to other situations is limited only by the creativity of the teacher and/or the students. Logo can be made simple enough for slow learners and yet it has the capacity to excite gifted students. Computer-controlled toys, such as the Logo floor turtle and Big Trac, provide a more tangible reflection of pupil problem solving techniques.

Simulations

Simulations also involve problem solving skills. Users are asked to assume a role and to make all decisions inherent in that role. Simulations place students in situations which they would not otherwise be able to experience - piloting an aircraft in Flight Simulator or mixing dangerous chemicals for a chemistry experiment, for instance. Gifted students find this application particularly challenging, as simulations provide a great deal of latitude for creativity and research. Handicapped pupils, who often have less opportunity than other students to explore their environment, can use programs such as Lemonade to develop or strengthen knowledge. Many skills are called upon by this application of the computer, for example reading, mathematics, problem solving and value judgments. Special students, working in groups, learn from each other and with each other as they explore exciting environments created by simulation programs.

Art and Music

Computer art and computer music have been vaunted as the "better" way for students to develop their creativity in these subjects. Indeed, these applications do harness some of the most striking and powerful capabilities of the microcomputer; but special educators should consider computer art and music programs as merely another medium to explore, in addition to current artistic and musical activities. The Koala Pad, and similar software, gives users the ability to manipulate colour, shape, and texture in a unique manner. Motorically limited students, who have never been capable of producing art independently, are now able to do so using the computer and appropriate input devices. Music software such as The Music Maker, facilitates exploration of music theory and the sounds of the various instruments. Students can use the microcomputer to create music and then to play their creation in many different ways. Students who are unable to play musical instruments because of lack of experience or lack of motor ability can experiment with the sounds of a variety of instruments, by using microcomputer technology. Computer art and music can be used to inspire creative writing, to illustrate stories or poems, to follow-up field trip experiences, and to expose pupils to the capabilities of computers, as part of a computer literacy curriculum.

B. Computers as Prostheses

To this point we have discussed how utilization of the computer can enhance learning in the special education environment. The microcomputer is also able to provide assistance in alleviating the discrepancy that often exists between a disabled student's ability and the performance of cognitive tasks. Many handicapped pupils are able to participate in the educational process with their non-disabled peers because of microcomputer technology. Special input methods such as single switch scanning, Morse code, oversize or undersize keyboards and voice or Braille input, allow handicapped students to access the computer and produce written output with the aid of word processing. Specialized software is available which facilitates mathematical computations, allowing users unable to hold a pencil or an eraser to regroup, erase, or work in long division.

Computers are able to output information in many different forms for example, Braille, Blissymbols, and synthesized speech, and can thus be useful as oral and written communication devices. Word processing, art, and music capabilities greatly increase the ability of disabled students to participate effectively in school. Two excellent resources in the field of assistive devices are the Hugh MacMillan Medical Centre (Ontario Crippled Children's Centre) in Toronto and The Credit Valley Treatment Centre in Mississauga.

C. Computers and People

Social Skills

Special students frequently display weak social skills. Computers utilized as part of the classroom programme, may facilitate the extension of interaction between special students and their peers. The lure of the computer encourages group participation which often heightens social expertise. Students who are talented in a particular aspect of computer technology, such as graphic design, may receive recognition from their peers, thus strengthening their self-concept (Turkle, 1984). One prerequisite to being accepted by one's peers is to be aware of the current trends in society. Computer expertise, or at least computer awareness, will help keep special students from falling farther behind their peers in social development Special youngsters often enjoy introducing some aspect of the computer to younger learners. This responsibility contributes to social and emotional growth. Computers are frequently employed most effectively by groups of two or three pupils sharing, compromising and learning together.

Software Evaluation

There are a few basic criteria to consider when evaluating software. A good piece of software - has no technical problems or "bugs"; is free of race or gender bias; reinforces correct responses with interesting feedback (but does not reinforce incorrect answers); is free of judgmental or insulting remarks; has appropriate sound features which are clear, unobtrusive and may be turned off; permits teacher input and modification; contains accurate, concise instructions; and has an instructional guide.

It is important to keep in mind the particular group of students for whom one is selecting software. A program which is inappropriate for gifted children may be very useful for slow learners. It is not necessary to limit oneself to software designed specifically for special education. Any program meeting the above criteria, in which language and content seem suitable, should be considered.

Help!

Contrary to popular opinion, one does not have to be adept at programming or be intimately familiar with the internal anatomy of the microcomputer to use it effectively in the classroom. A few hours of hands-on experience running programs and becoming comfortable with the computer, with an encouraging colleague nearby, is all that is required to get started. Try to think of the computer as another valuable teaching tool to be used throughout your programme, not as a new subject to be taught.

Teachers of average or gifted young people often have student computer experts to rely upon for assistance because, in many cases, students are far ahead of their teachers in computer knowledge. Teachers of slow learners or handicapped individuals have a particular responsibility to become comfortable with computer technology, as their students will be unlikely to develop computer skills without assistance. Lack of access to microcomputers may widen the gap between these students and their non-handicapped peers (Semmel, 1984).

A microcomputer in the special education classroom can encourage writing, increase students' pride in their work and in themselves, add a new dimension to the creative arts, facilitate problem solving, group interaction and logical thinking, and provide a powerful incentive for students to read. Computers are a part of our future. Computer experience should be an integral part of the education of our students.

References

Goldenberg, E. Paul." Computers in the Special Education Classroom: What Do We Need, and Why Don't We Have Any?" In Perspectives in Mental Retardation. J. Mulick, ed. Ablex Press, 1984.

Goldenberg, E. Paul et al. Computers, Education and Special Needs. Don Mills, Ontario: Addison-Wesley, 1984.

Hagen, Dolores. Microcomputer Resource Book for Special Education. Reston, Virginia: Reston Publishing, 1984.

Papert, Seymour. Mindstorms. New York: Basic Books, 1980.

Semmel, Melvyn et al. "Training Special Education Personnel for Effective Use of Microcomputer Technology: Critical Needs and Directions". In Microcomputers and Exceptional Children. R. Bennett, C. Maher, eds. New York: The Haworth Press, 1984, pp 63-81.

Turkle, Sherry. The Second Self: Computers and the Human Spirit. New York: Simon & Schuster, 1984.

Wanderman, Richard. "Bridge to Clarity: The Computer as Compensatory Writing Tool". In Softalk for the IBM Personal Computer, June, 1984 , pp 25-30.

Word Processors Or Story Processors?
(Redefining Literacy)

Glen R. Turnbull
Perth County Board of Education
5 Douglas St.
Stratford, Ontario

With the new advancements that are being made with computer technology and word processing software, teachers are faced with the challenge of developing innovative techniques in relation to the teaching of writing skills. To accept this challenge educators must be prepared to abandon some of the more traditional approaches and then attempt to gain insight into new philosophies and pedagogies as they relate to the writing of today.

Traditionally, we as educators have taught children the process of writing under very specific guidelines. These guidelines served a functional approach in acquiring a product from students' efforts in writing. Educators such as Donald Graves, Marjorie Franks, Joanne Yatvin and others, prescribed a number of stages that students were to be subjected to when experiencing any form of writing. Basically, what they were advocating was a composing process which led the student writer through stages of planning, writing, and revising in a very sequential manner. The revision stage seemed to be primarily limited to corrections in spelling and punctuation with changes in wording or phrasing on a limited basis.

These stages of writing suited the sequential manner in which stories were mechanically reproduced from the student's imagination onto paper via pencil or pen. Ideas flowed onto the page from beginning to end throughout the writing process. When it came time for revision, emphasis was primarily placed on correct conventions of spelling and grammar. So, as a result of this, students would be required to re-write the complete story to acquire a more presentable product. After several revisions at this level, the enthusiasm for the story would quickly diminish. Students learned that to avoid such aversive stimulation, one only had to decrease the total output of writing. This gave rise to one of literature's greatest paradoxes. In an attempt to improve student writing skills, a great deal of emphasis was placed on the writing process and acceptable conventions. However, the result was a dramatic drop in product. What we were accomplishing was the production of technically precise writing of inconsequential proportions.

It is little wonder teachers have found it so difficult to motivate children into experiencing the writing process. These children learned that the joy of expressing their thoughts through the written media was so short lived and would eventually lead itself to the tedium of recopying "ad infinitum." Cries of; "Do I have to do it over again?", and "Why do I have to write a story?", echo through our schools. What these children are really saying provides opportunities for the writer to move sentences and paragraphs, delete inappropriate text, insert new material or concentrate on the organization of the whole written text. In the past the act of recopying often discouraged large revisions. Now the process has become one of reorganization rather than recopying.

More important than the microcomputer itself, is the type of software that is being introduced to assist in the writing process. There are basically four major types of software that are currently being used in educational settings. The simple word processing packages, the Computer Assisted Instruction (C.A.I.) packages, secondary type programs that allow for corrections of spelling and conventions in writing, and Story Processors. Each of these reflects a philosophy of story writing and each provides some developmental benefits. However, there are also associated drawbacks which are dependent on the way each is utilized by various instructors.

Many of these programs do not indicate the existence of new teaching procedures but they do illustrate that the computer can imitate a form of instruction which is made available to the individual student. I must emphasize a word of caution at this point. The role of the teacher in this whole process must not be undermined in any way by these particular programs. There is a saying that computer programmers use when relating to writing BASIC programs which states "Garbage in; garbage out." The same reference can be made towards the use of any of these particular programs. There is no safeguard here that will screen the quality of input that a user may choose to enter as a story. These programs can not make a value judgement on the content of the story being developed. If the student is putting in total nonsense, then the final product will be the sum total of that nonsense, no matter how grammatically or stylistically correct. The teachers role here is a very crucial aspect of the whole writing process. Teacher interaction is imperative when attention is focused onto the act of creativity.

The process of writing has undergone many changes throughout its historical development. Educators "have learned now that classroom writing instruction in order to nourish the writing process, must be devoted largely to two kinds of activities: (1) getting ready to write the next piece, what some call pre-writing activities, and (2) responding in a variety of ways to (the) writing students produce, responding to essential matters, and responding in a way that keeps the flow coming, that keeps expressiveness and willingness high."(Cooper,1975).

Just as concepts related to reading and writing changed drastically with the invention of the Gutenberg press, we are on the verge of experiencing another monumental change in concepts with the introduction of microcomputer technology.

We are now heading into an era where everyone will have access to an avenue of creative expression: the microcomputer. Not only will individuals be able to express themselves in print, but this expression will be taking on the old traditional form established by the conventions of correct spelling, syntax and grammar. There is however, a big difference with the process today. Instead of these conventions restricting is, "I don't see the point in all of this." They have lost their creative energy. The surge of "idea abundancy" wanes as their story reaches its conclusion and very quickly dissipates when demands for re-writing are even suggested by a teacher.

Examine the following analogy. An artist paints a magnificent picture of a landscape. The scene is vividly expressed, as in its natural beauty. The artist is bursting with creative genius in trying to relate his feelings about nature to those who will enjoy his painting. Upon closer scrutiny, an art critic suggests that the oak tree in the right foreground should have a few more limbs and the pink of the sunset should be a little more washed out to suggest the end of a long exhausting day. The artist is now asked to turn his attention to a second easel nearby, bearing a blank canvas. He is then instructed to start over again by painting his original picture and to include the suggested modifications in the second painting. This is what educators are asking children to do with their writing. We ignore the initial creative burst of energy and place an emphasis on some of the more mundane details of correct form and acceptable standards. Children write to express their creative selves. We must emphasize this creativity within their writing and yet, still encourage effective literary models of communication.

Educators have found this virtually impossible to accomplish under the traditional means of story writing (ie. paper/pencil).

Ironically enough, (compared to the way some educators have taught writing in the past) it has been shown that experienced writers "revise early drafts structurally and conceptually and modify individual words and sentences in later drafts ... the goals of this process are to explore and clarify ideas and to create a written text that communicates effectively with the intended audience." (Kane, 1983). With the advent of microcomputer technology and the use of word processing, we as educators are presented with an opportunity to conceptually change our beliefs in relation to the story writing process. This process (the clarification and exploration of ideas) can be facilitated through the use of word processing and microcomputers. The original canvas is still there and all the child has to do is to add a little colour here and a few strokes there in order to enhance what he has already created. Eventually, every child will be capable of painting a story of fine literary quality and creative beauty.

Another revolutionary effect of computer technology on education is evident in the transition of two conditions related to the way English teachers will teach writing (Whitney,1983). Educators can now willingly abandon the strict teaching of spelling and grammar as being primary goals of writing. Teachers can now let the microcomputer focus on the correction of mechanical errors and thereby free the students so that they may concentrate on the development of different cognitive skills that are required to produce and organize story material.

The second major change that educators are experiencing is related to the introduction of the microcomputer itself along with all of the features it has to offer in aiding the process of writing. This new technology what is to be printed, the material that is being printed will be enhanced by correct conventions, the inclusion of which are facilitated by many of the existing software packages. This will certainly have a dramatic influence over not only writing, but also reading in our educational system. Gone are the "See Dick run." forms of writing. We will now, "Observe the agile Richard circumnavigate." Well maybe not quite to that extent, but changes are happening and at an alarming rate.

We now realize that writing is a process. It is a series of continuous actions that bring about a certain result. Many of these actions occur simultaneously. Often planning, writing, and revising occur all at once. Our role as educators will be to develop ways of teaching that will satisfy the needs of our students and utilize the full potential of the technology that is developing at the same time. The current movement seems to be away from linear patterns of expression and/or thought. Educators must now encourage students to work on the depth of their ideas as well as the latitude involved. This can be accomplished by replacing the phrase, "Write me a story." with "Paint me a story."

REFERENCES

Cooper, C.R. "Responding To Student Writing", The Writing Process of Students, 1975, No.1.

Kane J.H. "Computers for Composing", Paper presented at the Annual Meeting of the American Educational Research Association, Montreal, April, 1983.

Whitney, M.M. "The Computer and Writing", English Journal, 1983, Nov., pp24-31.

Using Microcomputers To Improve Self Image Of Learning Disabled Students

Brian Speed
Halton Board of Education
Oakville, Ontario

At Brookville, a K to 8 school in Halton County, microcomputers and the Adlerian Democratic Classroom are being used together to enhance the self image of Junior and Intermediate learning disabled special education students. The well known labelling problems, brought on by withdrawing students, have been effectively countered by using the special education resource area as a computer centre and by teaching special education students how to train regular students in the use of computers.

Background and Philosophy

The purpose of this article is to acquaint the reader with some of the details of Brookville's work with computers in the field of special education. Computers are being used there as part of a comprehensive plan to improve the image and self image of the special ed student. It was found that one highly effective way of introducing the microcomputer into a school is to use the special education room as the computer centre.

Brookville's Junior Intermediate special education programme is a withdrawal programme: most pupils take Math and/or Language in the special education class and then return to their home class to take other subjects. Approximately twenty learning disabled students work within individualized Math and Language programmes which make use of one to twenty (depending on availability) microcomputers at a time.

Prior to the introduction of the microcomputer at Brookville four years ago, the special ed students were outsiders. They risked being labelled and ostracized every time they were seen to go into the special education resource room. After implementation of the new programme, positive changes in the self image and attitude of the special ed students were noted. Behaviour problems virtually disappeared. Students who had previously refused to work began asking for homework. Students with short attention spans began working intensively, in some instances for periods of over an hour. Students began to ask for harder reading material. The special ed classroom became a place where everyone wanted to be. The Learning Disabled student now carried the label of 'Computer User' and was thus treated with respect.

When dealing with self image it is essential to recognize both subjective and objective frames of reference. Subjectively the student must come to see himself as a worthwhile human being. But this cannot be done if all his peers are constantly negating this view of self. Changes in self concept do not occur immediately, nor in isolation. We must give the students a healthy framework, a way of looking at life, into which they can fit positively. They must learn both to be responsible and to be seen as responsible; they must see themselves as worthwhile and be seen as worthwhile. What people think of us plays an important part in what we think of ourselves.

The microcomputer can act as a catalyst in the early stages of initiating a classroom programme aimed at enhancing self image. Its usefulness lies in the knowledge that the microcomputer is intrinsically interesting to most students. Thus, students using the microcomputer are often seen to be 'lucky', and students adept at programming and routine computer applications are generally regarded as 'intelligent'. Being lucky and intelligent are certainly new labels for the learning disabled student!

The special ed (withdrawal) classroom can also be a good area to begin microcomputer use for other reasons. First, because there are fewer students in the classroom, it is easier to make use of the computers on a regular, structured, continuing basis. By structured, we simply refer to an ongoing program that has some continuity from day to day. In addition the special education student generally requires more drill and practice than other students. The microcomputer will patiently generate an endless variety of questions (tailored to teacher specifications). This not only results in the saving of teacher preparation time, but produces a more enjoyable lesson for the student.

Planning for Success

Of course, more than microcomputers are needed to ensure the success of the programme: a philosophy is necessary to nurture and perpetuate the newfound selfworth the students find. At Brookville things are brought together by using the Adlerian Democratic Classroom model.[1] To the extent possible, students are placed in control of their lives by taking an active part in what goes on in the classroom. We give them responsibility in exchange for commitment and cooperation.

It is also important to have a plan. Here are some of the things you can do to get things started:

1. Talk to the grade 8 (or senior) students — their opinion can sway the school. Gain their support for the total programme. This includes refraining from teasing and 'putting down' the LD student population should this be a problem in your school. Note that once the programme is ongoing this problem will disappear because the special education students will earn and maintain the respect of the other students.

2. Talk to the special ed students too. Talk honestly about their disabilities and reasons for being in the class, and talk about your hopes, plans and expectations. The first day of the term is a good time for the first class discussion.

3. Employ class discussions regularly.

4. Ensure the special education room is the place to be by making the room the home base for the school's computers. Have the computer club meet in the special ed room every lunch hour. Encourage your special ed students to eat their lunch in the computer room if you have a bused school.

   [In the long run it may not be possible to make the special education room the computer centre, but that is not necessarily a problem. There is a marked decline in the fascination computers hold for students after a while anyway. What is important is that initially the computers are used to help the special education students achieve a new status within the school community. The focus then changes (no matter where the computers are located) to one of maintaining this new status.]

5. Begin to teach word processing (and even programming) to those special ed students who show interest in this topic.

6. Have special ed kids teach other students.

---

[1] For more information concerning the Democratic Classroom refer to Dreikurs: Maintaining Sanity in the Classroom.

7. Teach computer enrichment in the special ed resource room. (Keep the room active and encourage a variety of students to use the room.)

8. Have enrichment students work independently in the special ed resource room in addition to the library.

9. Invite students from other classes to spend their free time in the computer room.

10. Share information such as publications by the ACLD with your students on an ongoing basis. The effect of this is to make the students feel their disabilities are normal -- after all, a great many people are learning disabled.

11. Form a special ed baseball team (or engage in other kinds of group consolidation activities) and play other classes — borrow a good player or two if necessary to even the odds.

12. Make sure the special ed room 'shows off' the students. Post students' work. Take each student's picture and post it too. Display interesting and special projects in the room.

13. Give responsibility. Post it or write it out. This is important for self image. It is necessary that exceptional students be involved and be successful in that involvement. Social development is critical. Many LD students are not full participants in shared reality. They are on the outside looking in. Give them a 'computer role'. Have them demonstrate, deliver, set up and load the computers. Give them other roles. Have them sell popcorn, get a class mascot and take on the responsibility of looking after it. Send representatives to the student council. Have your own 'special education and friends' trip. Self image requires maintenance. These students are intelligent and aware. They must experience success in legitimate enterprises that can be validated by other students, teachers and parents in the school. In other words, the responsibility they are given must be 'real' and not contrived.

14. Get together with other schools in your area that are computerized in special education and initiate contact using a modem. Set up a Pen Pal system using a local Bulletin Board as a message depot.

Note the affective nature of the above items. The indirect uses of computers are probably even more important than direct uses. We can see powerful possibilities in terms of enhancing self esteem, of diagnosing and prescribing. But we do not disregard direct use. Not at all. Now that we have a philosophical framework, we examine the practical uses of computers. Here are some specific things they can do for your students:

- motivate
- establish computer literacy
- provide rewards
- promote social and personal skills by helping develop tolerance of others
- encourage sharing
- assist the student in making friends
- enable the student to perceive competition less threateningly
- interact by teaching others
- aid eye-hand coordination
- aid in the focus of attention
- foster confidence
- help give success
- help with practice, drill, and skill consolidation in Mathematics, Language and other areas as required.
- assist in the development of writing skills (word processing)
- improve thinking skills
- give immediate feedback
- help individualize programmes
- teach programming
- improve the appearance of written work
- check spelling
- act as a calculator

Implementing the Programme

After becoming familiar with some of the things a microcomputer can do, you can select key objectives. These specific goals will be of great help while working on the integration of the computer into various school programmes. A beginning computer user might create a few objectives such as indicated in the example below.

SAMPLE OBJECTIVES MICROCOMPUTER USE

The microcomputer will be used (as appropriate):

- to teach the fundamentals of computer operation
- to encourage students to write
- to motivate
- to enable students to focus on spelling and editing, to help them write longer stories while moving from rough to final with less writing and to reduce the incidence of reversals in their written work, through the use of Word Processing and Spelling Checking programs
- to provide drill, practice, and feedback consistently at the students independent working level (including the immediate consolidation of lessons)
- to teach or reteach specific algorithms
- to give the student regular success
- to help develop a positive self image and sense of responsibility by delivering computers to other classrooms, loading programs, and demonstrating the use of programs
- to improve the legibility of written work
- to help the student take mistakes less personally
- to provide variety in the programme
- to help promote logical thinking, ordering, deductive logic, spatial concepts, mapping skills, organizational skills etc. through the use of Problem Solving programs
- to promote social skills by working together (two students can work well on one computer) and by teaching other students (many special ed students work in the computer club during lunch as student teachers)
- to teach BASIC programming (specific programming objectives would also be required)
- to help the student master mathematical algorithms by practising using programs that take the student through a given process step by step (such as Adding, Subtracting, Multiplying, Dividing, Counting, Place Value, Fraction Operations, Decimal Operations)

Sharing Computers

As mentioned earlier, the resource room has the advantage of containing fewer students than the regular classroom. Students in this setting may thus be allocated computer time more easily than in the regular classroom. Not all students use the computer every day. Longer blocks of time seem to work most effectively. Computer assignment should be based on each student's needs and learning style. Some students (though few) do not learn effectively on the microcomputer, just as some students do not learn well by listening.

When setting up a computer programme the computers can be organized so that their home base is the special education resource room. All computers can be placed on carts so they may easily be moved to other classrooms. In this way we can more effectively serve all our students.

REFERENCES

Dreikurs, Rudolf; Grunwald, Bernice B.; Pepper, Floy C. Maintaining Sanity in the Classroom, Har-Row, 1982.

Dreikers, Rudolf, Discipline Without Tears, Hawthorn Dutton, 1974

On-Line Needs Diagnosis

Penny McCahill
Lakeshore Collegiate Institute
Etobicoke Board of Education
Etobicoke, Ontario

The dimensions of learning span a broad continuum. How can individual cognitive needs and sensitive personal concerns be approached concurrently within a diverse language classroom? How can modern-day technology facilitate one's dealing with cool intellectual issues involving grammatical inaccuracies and at the same time, warm emotional matters stemming from personal experience? Can both dimensions be accommodated appropriately for all individuals? This paper discusses a possibility toward bridging the gap.

As language students learn to process information, that which they are dealing with forms the substance for at least two sorts of analyses. The words that they string together, whatever the fashion, can be considered in terms of linguistic competence, and they can be considered from the standpoint of the message. When the critic, be it the teacher or a fellow student, is seated side by side with the learner at a computer keyboard, and when the text being evaluated has been set down using a simple word processing program, then relevant comments that apply to both the syntax and semantics of the text, can be easily applied in a constructive manner and learning can occur as an integral part of the writing process. Revision involving insertion, deletion, or reordering become simple accomplishments and can be dealt with in a relaxed, fluid manner. Language skills and skills of cooperative social interaction are developed concurrently.

This paper looks at authentic situations where "teamwork at the keyboard" played an important part in the learning process for secondary-level ESL students. An attempt has been made to point out ways in which the computer has been used to facilitate learning. Concerns raised during the process are also noted. In addition, the reader is introduced to a way in which the current use of the computer will be enhanced in this particular classroom in the near future.

The following excerpts are based on the efforts of intermediate and advanced-level ESL students at Lakeshore C.I. in Etobicoke, Ontario. Each student was in the midst of creating an autobiographical sketch. The project took place over eight weeks. The students were scheduled to work on their sketches every Thursday and Friday. In addition, several students chose to put extra effort into their writing outside the scheduled class time. All writing was done at the computer keyboard and saved on disk for ongoing revision. Some of the conferencing that surrounded the creative process took place at the keyboard during the writing process. Conferencing also occurred away from the computer on Wednesdays, sometimes individually, sometimes in small groups; hard copy printouts provided the substance for revision and the stimulus for grammar lessons for at this time.

Writing at both the intermediate and advanced levels was supported by the reading component of the courses. While the students were writing autobiographical sketches on Thursday and Friday, they spent Monday and Tuesday reading examples of the same discourse type. Explicit parallels between that which they were reading and that which they were trying to write were drawn. Most of these reading activities were done away from the computer; however, plans have been made to incorporate the exciting new program CLOZE ENCOUNTERS UNLIMITED into the reading section of each course in the near future. CEU is a piece of language arts software currently being developed that will stimulate language and thinking skills through computer-centred activities that involve the use of the students' knowledge of syntactic and semantic aspects of language.

And now to the products of this computer-centred writing project. Consider this first excerpt from the autobiographical sketch of a refugee from Saigon named Bill:

I was born in Vietnam on July 13/65 in the hospital. When I was one years old, I was so sick almost die. By this time my sister born, my mom told me maybe my sister took my milk then I didn't have enough milk and got sick. My dad busy his business didn't have to look after me, My neighbour helped my mom took me to hospital, we was unluck to have broken bed, the bed not made of cotton, it made of wood. The hospital not good like Canada. My mom didn't remember how the doctor saved me, my mom so glass I did survived, after I did survive but still have some problem, for example, little ear problem, speech problem, not having ability doing thing some people can do and language problem even in my own language, I just found out.

Where does one start to assist this student? There is no doubt that he could benefit from a conceptual understanding of English sentence structure. He needs to learn the difference between a comma and a period, and to come to an understanding of how and where to end complete thoughts. He needs to know that all English sentences have verbs. Semantically, he needs to learn an appropriate replacement for the non-word "unluck", and to sort out his phonological confusion with the words "glass" and "glad". And then there's the highly significant information regarding his particular learning problems "even in my own language". What does one deal with first? And how much should be discussed at one time?

Whatever the answers to these questions are, the computer will facilitate the decisions. It is possible to edit, print out, and edit again, as many times as it takes to deal with the problems in question. And after each editing conference, at the touch of a button, a clean copy can be generated for consideration at a later date. The student is only forced to deal with as much in the way of revision as he or she can handle at one time; and the tiresome job of manually rewriting the entire text each time is eliminated.

Realino's difficulties introduce another dimension important to language teachers. This young man from the Philippines independently decided to begin every sentence of his sketch on a new line:

I live in the Philippines with my father and mother and sister for fifteen years.

Then my older sister came to Canada afetr she married a Filipino guy.

Five years later my sister sponsored our family to came to Canada.

We arrived in Canada last November twenty eight nineteen eighty four.

My house in the Philippines is made of wood, and the ceiling made of nipa hat.

In the Philippines I have a dog name Barke.

Realino needs to be introduced to the notion of discourse structure. Where does one paragraph end and the next one begin? Does the fourth sentence connect more closely with the third or with the fifth, or is it irrelevant to the chain of thought altogether? How can the textual framework be rounded out with relevant details? And, once paragraphs have been set, what transition words and connectives are needed to establish cohesion? Once again, necessary revisions may be made easily and in manageable chunks because of the word pro capabilities.

A third student, twenty-year-old Iwanka, presents another challenge:

My parents were born in Ukraine and leter they were taken priseners into the Poland, and lived under the control of the police. The police were always checking our family, whotever we go they where after us. They also new where we went, to whom we tolk and meet. All this time we were saffer a lot of descremination, not onlly from the socialist police forse, but also from the Polish people. Whit our neighbours and our school friends, we had to pretend that we were Polish and not Ukrainens. But finally we moved and now we are in Canada where is a freedom.

Carl Rogers once wrote:

> When a facilitator creates, even to a modest degree, a classroom climate characterized by all that one can achieve of realness, prizing, and empathy; when he trusts the constructive tendency of the individual and the group; then he discovers that he has inaugurated an educational revolution. Learning of a different quality, proceeding at a different pace, with a greater degree of pervasiveness, occurs. Feelings --- positive, negative, confused --- become part of the classroom experience. (1967)

The writing of both Bill and Iwanka cried out for attention in a humanistic sense. In spite of all grammatical and spelling errors, there was no doubt as to their meaning. In light of their awareness of the opportunity for sharing though conferencing, they made thenselves vulnerable to the teacher and to their classmates. This trust in the overall constructive tendency of those involved made it possible for the teacher to more readily individualize instruction. In Iwanka's case, it was significant that she and her sister were the only two Ukranian students in the class of thirty; one third of their classmates was Polish. Iwanka's courage and willingness to share aroused a new degree of awareness and concern amongst her classmates. Mental blocks which might have interfered with learning were dealt with at the same time the language learning was taking place. In the same way, Bill helped his classmates to understand his learning problems and a heightened patience and increased willingness to help him were the results.

It is this writer's belief that the sharing took place at all because the work was on disk; it was easily called up and revised a little bit at a time as each writer got to know the teacher and the group situation better. Moods changed over the course of the project. Word processing capabilities allowed each writer to "open up" an already begun text in order to add a few relevant details that had been omitted at the time the text was originally created. As queries were made by those involved, changes were made with ease.

John Dewey stated:

> Education must be conceived as a continuing reconstruction of experience, that the process and goal of education are one and the same thing. (1897)

In this ESL classroom, using the computer as an interactive classroom tool facilitated to an infinite degree the process through which many worthwhile educational goals were practised and attained. In the affective sphere, through extensive interaction with classmates and the teacher at the computer keyboard, students learned to use cooperative social skills as they applied learned criteria and higher level learning strategies to texts that were to be evaluated. The students learned to deal with technology in a friendly controlled environment thus developing an attitude that expectantly will transfer outside the classroom. Through interaction with the computer itself, through extensive revision made possible through word processing capabilities, students developed their language proficiency. They consolidated and extended their knowledge of language structures and vocabulary through interactive, problem-solving activities in which they had personal involvement. They also developed an understanding of discourse structure and an awareness of conventional formatting. And repeatedly, through conferencing, they were encouraged to apply their knowledge and skills to the ongoing analysis and revision of their own writing and that of their classmates. Their willingness to interact and revise extensively was motivated and facilitated to a significant degree by the computer. The computer presence made learning of this different quality, proceeding at this different pace, and with this greater degree of pervasiveness, possible. The computer, when used in this interactive style, gave me, the teacher, the ability to map learning styles in a continuous manner, and at the same time made the learning process an experience in human relations.

References

Dewey, John. 1969. My Pedagogic Creed. Education Today. New York: Greenwood Press, P.12. Originally published in School Journal, LIV. January 16, 1897.

Rogers, Carl. 1967. The Interpersonal Relationship in the Facilitation of Learning. Humanizing Education, ed. R. Leeper. ASCD, NEA.

Appreciation is extended to the Graphic Arts Department at Lakeshore C.I., Etobicoke for allowing me to typeset this paper.

Some Thoughts On Data Management For Special Education

Joseph H. Benstein
Livonia Public Schools
15125 Farmington Rd.
Livonia, Michigan 48154

The use of the computer to contain, retrieve, and manipulate information provides for special education an ability to master the large amounts of information required by federal, state and local authorities.

Much repetitive paperwork can be introduced into computer storage and held until needed again. Such information as personal history, guardian information, IEPC dates, diagnostic data, scheduling information, historical records and progress reports can be retained by the computer. The question, therefore, becomes with so much informational capacity possible to individuals, why has there been a lag to computerize special education information. There are several reasons for this lag. Prominent among them are the following: cost, education, and politics.

Until recently cost has been a primary concern. Limited funds, without federal or state suppport has forced most school systems into careful deliberations deciding what areas should receive funding. As school systems shrink in pupil size special education funds are also diminishing. However, two factors make it now possible for computer purchases in special education. The first factor is the lower package price for a starter system. At present REIMC pricing places an Apple IIe or IIc, 128k, dual drive starter system under $1,300.00. The second factor is the state and federal budgeting agencies for special education have included computers as items that could be bought with special titled monies.

The second concern is the lack of knowledge and confidence to operate a data management system. Courses at colleges and universities have been geared towards degrees in system designs and programming functions, especially on mini or main frame computers. Recently, large acquisitions of microcomputers by universities such as Oakland and Michigan, and an increasing number of classes being directed at educators have increased the computer literacy function of teachers as a group. Still, special education data management remains a task that requires more than just computer literacy.

The third reason for the lag and probably the most prominent is political. Many systems have been reluctant to give the special education departments the "green light" to computerize their records. Some reasons that have ben cited are: Other more important departments such as business and personnel are in greater need. They should come first. The special education department, they say, is already over-funded. They don't need a computer. The ISD already sends them computer reports. Why do they need another computer? Their student population is smaller than departments in general education, shouldn't general education receive one first? These questions and others, while valid, are mainly politically motivated and do not address the issue of special education needs.

With the continued regulation and accountability by federal, state and local authorities special education finds itself inundated with paperwork. There is so much paperwork that many teachers feel the education of many children has become nothing more than a holding maneuver while teachers shuffle report after report. Accurate and concise record keeping is important in the case history of students.

Each of us, however, has had to continually reproduce paperwork. These pieces of information which are continually being demanded by the various agencies and departments can be computerized at a minimum cost to the district. Once the system is in place and has gone through a trial period of operation, less time can be spent on record keeping and more time spent on the management and instructional component of the individual child.

Several changes are occuring within the computer industry as well as within the system that makes the management of special education data possible. First, and probably the most important, is the technological breakthroughs that have happened in the computer industry. The increased capacity and speed of the micros along with improved software make it possible for large amounts of data storage with a relatively small amount of computer knowledge on the part of the operator. This move towards a more "user friendly" environment has taken some of the apprehensions away from operating a computer. Secondly, administrations have recently become aware of the computer as a teacher tool. This has happened mainly through the many professionals who have purchased their own equipment, and have aptly demonstrated the ability of the machine to manage information. More teachers are producing materials for their students, tracking attendance, and recording grades using a computer.

To begin the management of student records, we need to consider that following: staff involvement and commitment, teacher management of the project through leadership and committees, experimentation of different software packages, a trial period of testing and change, and a need to keep the tasks practical and simple.

First, there should be willingness on the part of the staff involved to become part of a commitment to the process. Without their support, no system will work properly. Such a case in point happened several years ago when the assistant director for special education decided that a new year-end report form was in order. When the form came out, many bitterly complained about the form. Filling out the form became a task without staff commitment. As a result, much of the information the assistant director wanted was either inaccurate or misleading. To this day, the form is filled out in a haphazard way. When working with the computer I am reminded by the adage "garbage in garbage out". The information we enter must be information that will be useful, not just to an administrator trying to complete some requirement, but to the teachers involved in the process.

Secondly, it is important for some person to take charge of the project. Having a special education teacher volunteer to learn, direct, and manage the system would help to insure teacher involvement.

Thirdly, that person plus the administrator involved in the project should enlist other interested special education teachers to become part of a committee to assist in the project. The makeup of the committee is important to insure the project's success. Success is dependent on the system being a teacher-initiated project.

Fourth, sufficient in-service time should be allowed for deciding on the system to be used and the appropriate software package(s). Such consideration as to type of operating system (DOS, MS-DOS, CP/M, PASCAL), specific hardware requirements (Hard Disk, 80 column board, Thunderlock +), and specific software (DB Master, Condor, Lotus 1-2-3, Sysdata) must be addressed before the start of any trial period. In fact, the decision concerning the software should come first. It will direct the type of system that must be purchased. When a software system is considered, someone should preview the materials first under actual conditions. Many of the materials, especially those designed for special education are considerable in price.

Fifth, once the system and software are in place a trial period, depending on the size of the information, usually one year, should be initiated. The trial period will be a time for experimenting, developing ideas and transferring these ideas to concrete form. During that period, a note of caution, do not change the present structure of operation towards computerized information. A duplicate process must always be in place. Power failures, machine malfunctions, incorrect directions by the manual, could all cause loss of information. It might be well to remember one of Murphy's Laws, "Whatever can go wrong usually will." A little caution at the early stages will save a lot of anxiety and frustrations later.

Further, always test the ideas out under actual conditions. Do not superimpose an artificial structure or environment into the record management. We will not know if the system will work as anticipated unless it is used by those involved. Lastly, be practical. Remember what Thoreau said, "Simplicity, simplicity, simplicity." You can always make changes later.

Postscript

For the past three years, we have been experimenting with record keeping systems. At present the special education department, Churchill High School, Livonia Public Schools is placing in computer memory the basic IEPC records, diagnostic and evaluative data, student schedules, student progress forms, performance objectives (language arts), and various classroom materials. This year, with the cooperation and assistance of the intermediate school district, the administrative services department is now placing registry information, bill-backs for special center programs, B1/B2 D-Day counts, personnel senority and certification records plus numerous other tasks into the computer's memory.

An Information Age Plan For Special Educators: Computers And Technology

Florence M. Tabor, Ed. D.
MCE Inc.
157 Kalamazoo Mall, #250
Kalamazoo, MI 49007

Special educators must plan ahead for implementation of technology into special education. With a coordinated well thought-out plan, computers and other forms of technology can assist the handicapped to control environments and to communicate effectively.

The changes exponentially occurring in our society are rapidly carrying us from a segmented assembly line Industrial Society to an individualized Information Age. The implications for our educational systems are tremendous, are BECOMING megadirectional, are to a large extent unknown, but are nevertheless real and cannot be stopped. This inevitable reality must be faced, now, by educators if our young people are to be ready. This inevitable reality must be faced by special educators because the implications for handicapped are like a two-edged sword. One side is faced with hope to provide technological equalizers for handicapped to control their environments, to communicate, to learn, to work, and to enjoy various leisure time activities. The other side of the sword provides the paradox. Robots will be, are being, programmed to take over many lower level jobs and, indeed, higher level jobs involving processes using artificial intelligence. Whereas, many handicapped in the past could be trained for jobs that would last a lifetime, now job skills change periodically and many jobs become non-existent within a very few years. Individuals must be able to transfer jobs many times during their working careers -- therefore requiring process thinking and skill transference plus retraining.

Special educators must change; they must be ready for tomorrow today; they must provide an educational plan that takes technology, often from near ground-zero, to a total coordinated lifelong curriculum to make the best uses of technology as it is developed for the betterment of our handicapped populations.

The classroom of the future will no doubt not be a "classroom" as we know it today. There may be some on-campus work involving interactive video; writing using a word processor; learning how to use technology as a vocational, educational, utilitarian, and recreational tool; instructing using two-way telecommunications involving cable or satellite; and/or engaging in small group discussions. There will be learning directly from the home interacting with teachers and other resources from anywhere in the world -- and perhaps from a source orbiting the earth.

The educator in this new Information Age will find definite role changes. There may still be some direct teaching but this role will involve more group leadership in discussions, individualized teaching, and counseling. The educator roles will involve more planning of individualized programs and designing of materials. The educator will evaluate materials for use and will design modifications of peripherals as necessary for individual handicaps and modify the accompanying software programs to operate these peripherals for both utilitarian as well as for instructional uses. It is not to be misconstrued that the educator will need to become a programmer, nor should be, but will require the skills necessary to communicate effectively with programmers and with electronic engineers.

The key to going from where we are to where we must quickly be is planning. The plan must be flexible in order to incorporate our students' yet undiscovered needs as we traverse from the known Industrial to the unknown Information Age. Therefore, encompassing an entire long range plan must be a comprehensive, well-organized formative evaluation system based on goals or standards to which progress can be measured. Any discrepancy will then need to be analyzed to determine if changes are necessary in order to meet goals or if it is the goals and standards that need attention (Provus, 1971).

The first step in the plan must be a thorough Needs Assessment to determine how the computer and other forms of technology can be used most cost effectively and produce positive educational results. These needs will include administrative, student management, and instructional uses. All three of these areas must be considered on a coordinated basis in order that data can be uploaded and downloaded as necessary using local networking and telecommunication systems.

Administrative uses include coordinating and tracking information for report generation, use of word processing for communication , and keeping current on information through electronic bulletin boards. Student management uses will include a total coordination of the Individual Education Plan (I.E.P.) process -- including assessment, error analyses and generation of the I.E.P. to tracking student progress and evaluating educational approaches and materials. Instructional uses of the microcomputer will not only include those rapidly becoming common today for drill and practice, concept development and tutorial instruction, simulation, and problem solving (both by interacting with programs and by programming using packages such as LOGO). Technology, in general, will also be used for not-yet-dreamed-of purposes to enhance the functioning of the handicapped to control their environments for learning, social purposes, operating utilities, work uses as well as for communication within and between environments. Already, the use of peripherals allows the handicapped and others to communicate and control environments through voice recognition, touch control, and even through thought processing. There are also peripherals that communicate to the handicapped and for the handicapped through braille, voice, print, and other appropriate outputs.

After the initial stages of the Needs Assessment, intensive Investigation begins using a team approach consisting of educators, business/industry, parents, and other members of the community. At this point a Perting, or graphic plan, is developed indicating what is to be done, how it is to be accomplished, and the timelines to follow in reaching these goals.

A major segment of this process will be a coordinated Inservice/Educational Program designed to bring the educator, the business person, the parent, and the general public from a relatively lack of understanding and knowledge base on the use of technology to a comprehensive realization of the possibilities for technology as equalizers and tools for the handicapped.

Will our special education systems be ready for the coming of the Information Age? This Age is coming and is coming now -- whether educators are ready or not.

Provus, M. Discrepancy Evaluation. Berkley CA: McCutchan, 1971.

Introducing Computers Into A T.M.R. School: A Proposal

Stewart Robinson
Applewood Acres Public School
2180 Harvest Drive
Mississauga, Ontario

In the paper strategies for introducing computors to both the teaching staff and students in a school for the trainable retarded are proposed.

With the introduction of computers into a TMR school, teachers face brand new curriculum challenges, for which there is little experience or no available data upon which to make decisions. At the heart of this question is a computer and as we move to the ideal of a "computer experience" for each student the questions of "why" and "what for" must be addressed.

I believe that there is an unlimited future for computers in schools. They can become a teaching aide, capable of individualizing and personalizing instruction. They can help with management and accountability services for teachers swamped with paperwork. They can provide instructional simulations that have been in the past too costly or impossible to achieve and they can become a conceptual tool for students who have been taught to program the computer to solve problems. Only by becoming computer literate will both teachers and students be able to turn this promise and potential into reality. The answer to "why" and "what" will be obvious when a computer in a classroom helps teachers and students to do the work of learning.

WHAT IS COMPUTER LITERACY?

Harper and Stewart point out that "the term computer literacy has become a buzz word that has taken on a range of meanings. To some teachers, literacy program means a concentrationed course in Basic programming: other teachers cover topics in "computer awareness", offering little or no programming experience at all. It is generally felt, however that everyone should have some knowledge of what computers do, how they do it, and what their implications are for us all.[1]

HOW WOULD ONE ACCOMPLISH COMPUTER LITERACY?

Although there are many perceptions of computers, teachers must learn to use and control this medium. The computer can be seen as non-humanistic, too difficult to use by the average student, or only as a toy for games. There is some element of truth in all these perceptions but when the teacher learns to control, then that person will be able to recognize and shape the computer's positive aspects to meet the needs of both the teacher and the TMR student.

A. INSERVICE WORKSHOPS FOR TEACHERS.

Inservice workshops should be set up to give hands on experience to allow teachers to learn how to use and control the computer. Ideally such workshops should be limited to two teachers per computer.

1. STRATEGIES FOR INSERVICE WORKSHOPS.
   1) Success from the start.
      -quality pieces of software would be run whose prompts and clues are well written to ensure that the user can proceed and answer questions without confusion.
   2) Proceed slowly.
      -teachers must understand and become successful users, whatever time it takes.
   3) Getting help from one's friends.
      -working in a team of two helps to humanize the experience.
   4) Mistakes can be corrected.
      -language with vocabulary and syntax is precise.
      -there is however no record of mistakes.
   5) Faith in the machine is faith in yourself.
      -when a person understands what it is that makes the computer do what it does, only then can that person control it.

CONTENT OF INSERVICE WORKSHOPS.

It should be noted that it is not the intent to produce programmers capable of producing instructional software but simply understanding and control. The content is divided into three levels:

1) Awareness
   -what a computer can do.
   -history

2) Orientation
   -concepts
   -parts
   -uses
   -limitations
   -access

3) Programming
   -BASIC
   -SYSTEM - NEW
   -RUN
   -LIST
   -PROGRAM -Print
      -Goto
      -let
      -input
      -dim
      -for next
      -read....data
      -graphics
      -plot

B. PROGRAM FOR STUDENTS.

The teaching strategies will be the same as those used for the inservice workshop. The content of this program is written in three functioning levels. Students will progress at their own rate through the different functional groups.

LEVEL I

Knowledge of the use of computer hardware with hands on experience.
a) Students will be able to turn equipment on/off.
b) Students will be able to load software into computer system.

Knowledge of the use of computer software.
a) Student will be able to interact successfully with a variety of programs.

LEVEL II

Knowledge of the difference between hardware and software.
a) Students will be able to successfully identify computer hardware.
b) Students will be able to demonstrate an understanding of software.

Knowledge of the use of computer hardware including hands on experience.
a) Students will be aware of minimal trouble shooting techniques.
b) Students will be able to demonstrate proficiency with the key board.
c) Students will be able to

demonstrate the use of storage devices.

Knowledge of career opportunities involving computer technology.

a) Students will be able to cite examples of computer applications in business, industry, and education.

LEVEL III

Knowledge of the use of the difference between hardware and software.

a) Students will be able to demonstrate an understanding of how data is stored in a computer.

Knowledge of the use of computer software.

a) Students will be able to make minor modifications of existing programs.

Knowledge of the social and ethical implications of computer technology.

a) Students will be able to describe both beneficial and harmful effects of computer applications, i.e. problems of privacy, computer crime, etc.

Ability to create successful programs appropriate to the students grade level.

COMMENT:

I believe that when children learn programming they are learning high level thinking. When a student learns how to use a computer they may be learning more about thinking than has ever been possible before. Papert[2] has pointed out in his book Mindstorms that in teaching the computer how to think children embark on an exploration about how they themselves think.

References

1) Harper, Denis and James Stewart, 1983, Run: Computer Education, Brooks/Cole publishing comp., California, p.24.

2) Papert, Seymour, 1980, Mindstorms, basic books inc., Publishers, New York, pp.230.

TEACHING

Looking At Learning From Lessonware

Ruth Pike
Faculty of Education, University of Toronto
371 Bloor Street, West
Toronto, M5S 2R7

The Exemplary Lessonware Project of the Ministry of Education, of Ontario involved the development and field testing of more than forty pieces of educational software. Lessonware projects covered all curriculum areas and grade levels from Kindergarten through Grade 13. A qualitative analysis of the field test reports identifies some of the dimensions of variation that need to be considered in evaluating the educational potential of software. These include variations related to software design, context of use and user characteristics.

We have remarkably little empirical data to guide us in harnessing computer technology for classroom use. Most of the literature on educational applications of computing are based on rhetoric, not reality. Classroom experience does not tell us much because the use of computers as an instructional tool is still very limited (e.g. Shavelson et al, 1984). Educational effects experiments tend to deal with computer instruction per se, to ignore variations in software and its use (e.g. Kulik, Bongert and Williams, 1983). There is a rapidly growing body of software reviews but these reviews are descriptive with evaluations based on inference and intuition, seldom on field tests.

Yet in order to be able to design software that optimizes learning and to be able to make intelligent use of computers in classrooms, we need to understand the complex relationships between types of programs, users and usage. This paper describes a beginning attempt to delineate some of these relationships.

The Exemplary Lessonware Project

In 1982, the government of Ontario became concerned with encouraging the development of educational computer hardware and software. An early project, the Exemplary Lessonware Project of the Ministry of Education, Ontario, funded 57 pieces of lessonware, covering many areas of curriculum content and designed to be suitable for children from kindergarten to Grade 13. Part of the agreement was that each piece of lessonware be field tested. This field testing was to serve two purposes. It would supply formative input into the final design of each program and also provide some estimate of its pedagogical potential. Because of the way in which the project evolved, the developers of the lessonware were not given rigid specification for field testing. They were provided with some printed guidelines (Pike, 1983) and consultation was available.

The field tests and the field test reports varied considerably. Some developers used sophisticated measurement instruments, prestests and control groups. Others relied on more or less unstructured observations. Most of them, however, provided some documentation of where and how students used a particular piece of lessonware and some of the things that happened as a result of this use.

The Exemplary Lessonware Project was not planned as a research project. However, we made a post hoc decision to make as much use of the experience as possible, to look at outcomes as if they could provide at least some tentative directions for future work.

Method

The method used is best described as a qualitative meta-analysis. Because it seemed intuitively unlikely that all types of lessonware would perform in the same way, a three-way classification system was developed. The system used those characteristics which were considered most likely to have some influence on learning outcomes. Each piece of lessonware was classified according to its intended instructional role, the nature of its learning objectives and its instructional mode or style. Some pieces of lessonware were placed in more than one category if that seemed appropriate. For example, one science program had didactic as well as problem-presentation elements, a game program presented factual knowledge as well as providing practice in problem-solving skills. The classification system served as an organizer for the subsequent analyses.

The field test reports were analysed collectively first, to identify any outcomes that appeared to have some degree of reliability. These included similar findings that were reported by several developers, the results of a few field tests that had been conducted and reported with some scientific rigour or my own interpretation of some sets of raw data that had been provided by the developers. This material was then reorganized and reexamined to see what sorts of patterns emerged, what sorts of insights were generated about the problem understudy.

Results

1. Variations in evaluations related to software design features.

   - Successful teaching was unrelated to subject matter content. There were more and less successful projects in all curriculum areas.

   - There were variations in learning outcomes associated with different categories of software. There was more success reported with programs used as reviews of content than with programs attempting to teach new content. Programs whose objectives involved facts, knowledge or concepts were perceived as more successful than programs concerned with skills. Problem presentations were not well rated, games which presented problems only slightly better. Examples of other instructional modes, i.e. simulations, microworlds or tools were effective in some circumstances.

   - Programs with multiple objectives sometimes had poor results because conditions favourable for the achievement of one objective might be counterproductive for another.

   - Conflicts between content objectives and format lead to poor learning outcomes. An example is a game which elicited rapid responses while the learning objective, solving word problems, required careful reading, reflective thinking.

   - Reading skills were a limiting factor even in secondary school programs. Both quantity and difficulty of on-screen text affected learning outcomes.

2. Variations in evaluations related to field test methods and context.

   - Students were more successful in achieving intended outcomes, programs were rated positively where teaching of prerequisite skills and knowledge that students lacked was part of the introduction to program use.

   - It is essential to distinguish between programs intended to introduce new knowledge or skills and those intended to reinforce what has already been learned. The actual instructional role may be different from the intended role if the program is not properly integrated with other classroom activities. The success of simulations was particularly context dependent.

- The evaluation may depend on the teacher's objectives for the activity. Some teachers treat computer experience as a cost-benefit trade-off against content-learning time and will tolerate programs that may be inefficient from an instructional perspective.

- Grouping of students may affect apparent success of programs. Students prefer working in pairs because the computer seems less intimidating and they think they learn faster. Paired students work longer than solitary students. Pairing also lead to positive social outcomes. There were several reports of increased co-operation, peer tutoring and so on.

- Support materials are an important part of the context of use of any given program. Both learning and attitude varied with variations in use of support material.

- Different data collection methods will elicit different kinds of information about any given program and lead to different evaluations. Teachers were sometimes satisfied if some individuals learned from a program even if group means showed no overall gain.

- Student evaluations of lessonware are strongly affected by individual learning outcomes and payoffs. Most criticisms of pedagogical qualities of programs came from students who found programs difficult or when content was unrelated to current courses of study.

- Students displayed personal preferences for design features compatible with different personality or learning styles. Passive students loved watching an animated graphics segment of a program. Other students were bored because the program proceeded in the absence of user input.

- Students were not reliable reporters about the presence or absence of objective features of the program like self-pacing. Evaluations depended on program satisfaction or responsiveness to their own needs.

- Student satisfaction with the sort of feedback provided depended both on their past experience and on the nature of the program. Only programs which did not resemble typical scholastic activities were acceptable when no explicit feedback to student responses was provided.

3. Variations in evaluations related to user characteristics.

   - The different modes were not equally effective for all students. The didactic mode tended to be more successful with secondary school students than with elementary school students.

   - Young children in kindergarten and primary grades used computers very differently than olders students. The computer hardware and the software program often served as catalysts for other activities rather than in their intended roles.

   - The pedagogical success of software was dependent on the entry skills of the users. Programs demanded prerequisite skills related to curriculum content, to reading level, to general thinking and learning skills or to skills specific to the instructions or the format.

- General ability level affected students success in learning. In some cases, slower children were able to benefit from the computerized instruction, in other instances they required much teacher support. In general, open-ended programs created fewer problems for less able children than programs with structure or defined objectives.

- Prerequisite content skills were more important in learning outcomes than computer experience. However, novice users had different attitudes about programs than experienced users.

- Girls are more cautious, hesitant to use the computer and required more instruction on didactic, problem and simulation programs than boys. The lack of ease with the computer did not, however, necessarily hinder learning.

- The appeal of particular formats varies to some extent with the sex of the user. More elementary girls than boys disliked an arcade type game format. A simulation game was more appealling to boys than to girls.

Discussion

Any interpretation of findings apply only to the set of computer programs examined, not to the entire universe of computer instructional materials. That is, just because some objectives were not achieved by a given subset of these programs does not mean that no programs of that type will be successful in the future.

The results are indeed modest. Neither the positive nor the negative rhetoric has been supported. We do not report any miracles for computer instruction. Some material that is hard to teach by other means was successfully taught. Some hard-to-reach students were reached. We cannot claim, however, to have made any fundamental changes in thinking skills, learning styles or general ability. (cf. Papert, 1980) On the other hand, no students appear to have been damaged, socially, cognitively or emotionally (cf. Hawkridge, 1982, Sardello, 1984). What the exercise has accomplished is the identification of some of the dimensions of variation that need to be taken into account in understanding the teaching potential of computers. The varying outcomes of the different types of software suggest that we need to search for information more specific than a general "computer effect". The differences introduced by the context of use reminds us that learning does not take place in a vacuum. Most importantly, the numerous examples of individual differences indicate that whatever it is that is transmitted by computer is not received by a passive black box. Even if this medium had only one message, the results would be as diverse as the students using it.

References

Hawkridge,D. New information technology in education. London: Croomhelm, 1983.

Kulik,J.A. Bongert, R.L. and Williams, G.W. Effects of computer-based teaching on secondary school students. Journal of Educational Psychology, 1983, 75 (1), 19-26.

Papert,S. Mindstorms: Children, Computers and Powerful Ideas New York: Basic Books, 1980.

Pike,R. Evaluating the Effectiveness of Lessonware Prototypes: Some Guidelines for Use in the Development of Educational Software. Toronto: Ministry of Education, Ontario, 1983.

Sardello,R.J., The Technological Threat to Education. Teachers College Record, 1984, 85 p. 631-639.

Problem Solving: A Computer Oriented View

David Moursund
University of Oregon
1787 Agate Street
Eugene, OR 97403

Among all creatures of the earth, humans appear to have the greatest ability to learn to solve problems. Problem solving is stressed in all of education; reading, writing, and arithmetic, as aids to problem solving, are considered basic survival skills. Now we have a new aid to problem solving--the computer. This paper explores educational uses of computers from a problem-solving viewpoint. The analysis reaffirms the importance of the basics, such as the three Rs. But it also suggests the need to integrate computer usage throughout the curriculum.

This presentation is based upon three basic observations.

1. Humans are problem solvers; the human brain and body are well suited to developing considerable ability to solve a wide variety of problems.
2. Humans have developed physical aids to help them solve problems.
   a. Aids to human physical strength include the train, automobile, airplane and other tools or machinery.
   b. Aids to the human sensory system include the telescope, microscope, radio receiver, and other such instruments.
3. Humans have developed aids to the intellect, such as reading, writing, and arithmetic (mathematics).

As one grows up in a particular environment, one develops the necessary basic survival skills for that environment. We learn by observing others and we learn by trial and error. But in our society that is not enough. Basic skills in reading, writing, and mathematics are so useful in solving everyday problems in our society that they are often called survival skills. Many of the tools and instruments we have developed are so complex that extensive formal education and supervised experience are needed to learn how to use them.

Added to this is that the accumulated knowledge of the human race continues to rapidly grow. We attempt to solve more and more complex problems by building upon this accumulated pool of knowledge--by building upon the work of others. A combination of relatively narrow subject matter specialization and working together in teams has allowed the successful completion of very difficult tasks, such as the moon expeditions.

Now a new aid to problem solving--the computer--has become commonplace. Computers can be built into existing tools and instruments, perhaps increasing their capability, reliability, or ease of use. Computers are an aid to the mind, a new aid to information processing. Not only do computers enhance and/or change the nature of the three Rs, computer and information science is an important subject in its own right. Most important, computers provide a new way to build upon the work of others. It may take only a few minutes to learn to use a computer program that took its authors thousands of hours to write. As educators, we need to provide leadership in assuring that the general population learns what it needs to learn about computers.

Let's begin with a quite general notion of problem solving. A problem consists of three distinct parts:

1. A given situation (initial state).
2. A desired situation (goal state).

3. Actions that can be taken that might transform the initial state into the goal state.

Often one encounters problem situations in which one or more of the three distinct parts of a problem are not clearly specified. Thus, an important aspect of problem solving is working to specify the three distinct parts of a problem. This tends to require a broad general education as well as knowledge of the particular problem area under consideration. If the initial state or goal state involves computers and related technology, then some computer knowledge is necessary merely to understand those aspects of the problem. Thus, one aspect of computer literacy is having enough general computer knowledge to be able to understand the initial and goal state of a reasonable range of computer-related problems.

Each discipline has developed its own techniques and tools for attacking problems. Computers have become accepted and useful tools in every academic discipline. For any particular problem, if a computer is possibly useful in moving from the initial state to the goal state, then computer knowledge may be useful to the problem solver. This suggests that one needs to integrate computer study and use into each academic discipline, and it provides another component of a definition of computer literacy

This superficial analysis can serve as a basis for developing a school's computer curriculum and for revising the non-computer curriculum. In this short presentation we can't go into too much detail, but we can look at some representative examples. In all of these examples we are assuming that students will continue to acquire basic skills in reading, writing, and mathematics.

To begin, there is the general idea of efficient use of the school time. We know that computers are useful in every academic discipline. Rather than have every teacher teach the use of computers, instead we need to sort out the basics and have them taught early on. For example, very young students can learn to access a computer system, and how to load and run a library program. This is an example of a basic computer skill.

The use of computers to store, process, transmit, and retrieve information is becoming commonplace. The initial problem is to communicate with a computer, in order to use a computer to help solve other problems. This suggests the need for input and output skills. Since we are assuming basic skills in reading, we assume students can read computer output. Someday we will have relatively inexpensive voice input devices. Perhaps voice, mouse, and two-fingered keyboarding skills will then suffice for computer input. But for many years to come, good keyboarding skills will be essential. By good, we mean the ability to keyboard at least as fast as one does handwriting, and using techniques that can be built upon to gain still greater speed and skill. Current evidence suggests that a fourth grade student can acquire the necessary keyboarding skills in nine weeks of instruction/practice, about thirty minutes per day. This is a basic computer skill that students should be acquiring.

The specific problems of storage, processing, transmitting and retrieving information each require training and experience. The amount of formal training varies, however. For example, suppose that one has a word processor with a spelling checker. If one knows how to read, write, and keyboard, it takes less than an hour to develop an initial level of useful skills in using the word processor and spelling checker. Of course, continued training and practice add depth, breadth and increased speed to one's skills. Note also that continued use of a spelling checker is a good way to improve one's spelling skills. This type of immediate feedback on errors in spelling the words one needs to use in writing is excellent.

As another example, consider the training and experience needed to search computerized data bases. Once again very minimal skills can be developed in short order. However, higher level skills depend not only on knowledge of the technology, but also on knowledge of the subject area in which one is retrieving information. Once again we see the need for a broad, general education. Since information retrieval is one of the fundamentals of a modern education, it seems clear that we should be providing substantial training and experience in both non-computerized and computerized information retrieval.

It is evident that computers are an everyday aid to solving problems in business, government, and industry. They are producing significant change in the work place. As young people go to school and gain in maturity, they can gradually accept more responsibility for making decisions about their own schooling and careers. They need general knowledge about how computers are affecting occupations to help them make appropriate decisions.

But here our school system also has a major responsibility. A high school student can see that computerized instrumentation is now a common part of new cars and of car repair shops. If the student decides to become an automobile mechanic, the courses available must incorporate appropriate instruction in electronics. A course in Pascal for college-bound students will not provide the type of training needed by the auto mechanic.

Perhaps the most challenging example is provided by mathematics education. If a real-world problem is to be solved by making use of mathematics, a five-step process is apt to be followed.

1. Understand the problem (using broad, general knowledge).
2. Represent the problem as a mathematics problem. This is a mathematical modeling task.
3. Solve the mathematics problem.
4. Translate the results back into the framework of the original real-world problem.
5. Check to see if the real-world problem is solved.

In most school systems, the greatest amount of mathematics education time is spent on step 3. But step 3. is the one that computers can do best. As a simple-minded example, consider basic computational skills. Nine year olds, given a calculator and a few minutes of instruction, perform as well on computational tests as thirteen year olds who are restricted to use of paper and pencil. This sort of observation has prompted the National Council of Teachers of Mathematics to recommend sweeping changes in the math curriculum. But schools have been very slow to respond!

Perhaps the single most important idea from the computer field is that of a computerizable effective procedure. The idea is to have students learn that a computer is a machine that can rapidly and accurately carry out the steps of certain types of procedures, and that computerizable procedures can solve or help solve a large and rapidly growing number of problems. A very strong argument for requiring all students to study some computer programming is that this will help them to understand the concept of a computerizable effective procedure. One learns quite a bit about this area by learning a programming language and attempting to write computer programs to solve a variety of problems. Note that the goal is not to solve the problems. Rather, the goal is to better understand what is involved in developing computerizable procedures to solve the problems. If one merely wants to solve the problems, likely there are library programs that can do this. This analysis suggests that such introductory computer programming courses should be taught by people who have a good understanding of developing and representing procedures to solve problems.

As a last example, I'd like to briefly discuss the idea of artificial intelligence (AI). There is a significant and growing range of problems in which artificially intelligent computer systems can perform at the level of a well-educated human expert. It is now clear that eventually AI will be a common aspect of computer systems. This lead to the following basic question. If a computer can solve or help solve a certain category of problem, what do we want people to learn about solving that category of problem? There is no universal answer--each general category of problem needs to be analyzed, and decisions needed to be based upon a good understanding of goals of education and the nature of our society.

One approach would be to have students involved in doing the analyses, so they can make appropriate decisions for themselves. That is, students should get an opportunity to study a number of situations in which AI systems exist. We can use a broad definition of AI, so that a spelling checker, grammar checker, graphic artist's package, chess playing program, and voice synthesizer all are included. Students can come to see machine skills versus people skills, how they compete and how they work together. My personal opinion is that students should learn to work with computers--neither being completely dependent on the machines nor striving to be completely independent. For most of us that has proven to be a successful approach to the other tools and instruments that humans have developed to aid them in solving problems.

It Figures: The Computing Component Of The Second International Mathematics Study

Wm. P. Goddard
University of British Columbia
2125 Main Mall
Vancouver, B.C.

ABSTRACT
Are you feeling logish? Do you suffer from sinusoidal irritation? Puzzled by the calculus? Sitting on your asymptotes? Fast, fast, fast relief.

During the 1980-81 school year, Grade 8 and Grade 12 & 13 mathematics students and their teachers from more than twenty countries participated in the Second International Mathematics Study sponsored by the International Association for the Evaluation of Educational Achievement (IEA). One of the objectives of the study was to compare the ways in which secondary school mathematics was being taught and, to the degree possible, to investigate relationships which might exist between differences in methodology and students' achievement in and attitude toward mathematics. Two populations of students and teachers were investigated, the first corresponding to the Grade 8 level and the second to Grade 12 or 13. These are referred to, respectively, as Populations A & B. At the Population B level, the examination of teaching processes was categorized as a national option for the International Study, and only British Columbia, Ontario, and the United States took part in that aspect of the project. Because of this, no analysis of this data was planned as part of the international reports of the project. It was the purpose of the present study, sponsored by the Social Sciences and Humanities Research Council of Canada, to conduct such analyses.

The richness of the data base which was developed is indicated by the following. The students of more than 450 teachers answered 84 questions about themselves, answered 17 test items each from a pool of 136 items, and indicated which items they felt they had an opportunity to learn. In addition to each student's success on an individual item and each student's individual test score, class average responses and scores were available for correlation with the answers which the classroom teachers gave to 17 questions about their schools, 51 questions about themselves, 152 questions about General Classroom Practices, 136 questions about their view of the student's Opportunity to Learn and more than 1390 questions on Topic Specific Questionnaires. The resulting three dimensional array of students, questions, and teachers provided more vectors than could be investigated in this study, but a beginning has been made. The present paper will look at a few aspects of this study related to the impact and use of computers and calculators.

FEELING LOGISH?
The topic of logarithmic functions is one of the most prevalent strands in college algebra at the Population B level. It was broken down into four subtopics; laws of logarithms, graphing logarithmic functions,

natural logarithms, and applications of logarithms. It is the last of these which we wish to treat here. But what is meant by 'applications'? The amount of time spent on applications of logarithms raises the question of whether this time is spent on logarithmic calculations (that is, on doing arithmetic by logarithms in this age of calculators and computers) or upon such applications as pH, normal curves, and exponential growth and decay problems. Only 47% of the teachers taught or reviewed formulae such as 'log(base b)x=y and log(base a)x=z if & only if b to power y = a to power x'. It is clear that any 'applications' do not have a very theoretical basis. This is our first clue as to what is happening.

Before the curricular reforms of the New Math era and the advent of calculators and computers, logarithms were studied primarily for computational applications. Logarithms were also frequently used to explain the basis for computation with a slide rule. Curriculum reform materials emphasized the importance of the properties of logarithmic functions. It is quite surprising, therefore, to find that 30% of the teachers spent more time on computation as compared with properties. The fact that 80% of the teachers reported that students generally used calculators in their classes makes it difficult to understand why the emphasis was placed upon logarithmic calculation and why, even in these calculations, only 49% of the teachers used calculators or computers in logarithm lessons . Forty percent of the teachers used logarithm tables. It is a relief to find that almost none used a slide rule, however, almost none used the computer either, although one student in six had access to a computer in mathematics classes in the US, one student in eight in BC GRADE 12, and one student in 50 in ONTARIO 12.

Other examples of antique calculation can be found in the college algebra portion of senior mathematics, for example, in solving systems of equations. Suffice it to say that there is no shortage of other material upon which the time could have been spent.

SINUSOIDAL IRRITATION?

Which topics from the trigonometry covered in our Grade 12 & 13 courses would lend themselves to the use of calculators and computers with a resulting saving in time and/or increase in understanding? One can mention the calculations necessary to graph periodic functions accurately and to solved triangles. But especially, one would think of the graphic capabilities of a computer to allow experimentation .
For example, a trivial computer program would permit students to input equations of the form:

$$y = a \sin bx$$
$$y = a \cos bx$$
$$y = a \sin bx + c \sin dx$$

for any values of a, b, c or d and see what happens. This could lead students to undertake the inverse process of decomposing functions. This area of harmonic analysis is one of the most important areas of research in modern mathematics and students, with a little help from a computer, could discover the living, exploratory nature of real mathematics. But who has time to do this in our present state? We are too occupied with calculating - with stuffing numbers into an algorithmic sausage machine and packaging the results as 'answers'. For performing trigonometric calculations the use of calculators having trig functions on them was only slightly more common than the use of four function calculators combined with trig tables.
The use of log tables alone was taught by 69% of the teachers. It is no wonder that Papert has labelled our courses as 'schoolmath' - a subject unrelated to mathematics.

Our calculations are not even realistic, practical applied mathematics. Even if the ability to churn out calculations were important, the calculations which students are able to perform are limited to 'cooked' examples with simple numbers.

PUZZLED BY THE CALCULUS?
The calculus referred to is not differential or integral. These are other topics we do not have time for in senior high school mathematics courses. Rather, the term 'calculus' is used here in its medical sense - a stone which blocks a passage. Our students, as was stated before, are blocked in their passage from computing (which the Oxford Dictionary defines as mere 'reckoning') to calculating (which the Oxford decribes as 'to ascertain by mathematics'). Although many Grade 12 students in the world (those with whom our students must compete in the future) have already proceeded to the calculus of mathematics, and to statistics, and to many other topics, we are still spending our time training students to imitate machines. Quite apart from considerations of the necessity in the modern world to be able to work with computers and calculators per se, we are substituting useless skills. We cannot even hide under the umbrella of 'pure mathematics'. The Second International Mathematics Study will show that we demonstrate very little proof and derivation of principles to our students, and we demand even less from them. What are we doing? Let us look at the topic of analytic geometry.

SITTING ON YOUR ASYMPTOTES?
It has been jocularly suggested that Descartes devised analytic Geometry over a week end. Be that as it may, the topic has attracted varying degrees of attention from teachers. For example, in BC GRADE 12 the time devoted to this topic ranges from 6 periods to 45 with an average of 21 periods. Seventy-one percent of all teachers in the TOTAL population taught the equations and graphs of conics. What about the other topics in analytic geometry? - the parametric equation of the line in a plane, the analytic proofs of geometric theorems, vector operations, translation and rotation of axes, the polar coordinate system, and the whole world of three dimensions?

Teachers were asked their reasons for including and omitting each subtopic. The prescription given in the course of studies or syllabus was the reason chosen by almost 60% of the TOTAL population - and was the reason chosen by an even higher proportion of Canadian teachers. The fault does not lie entirely with the classroom teacher!

The observations made previously about the use of the graphic capabilities of computers in trigonometry applies equally well to analytic geometry.

A student could be asked to input formulae of the form:
Ax(squared)+By(squared)+Cxy+Dx+Ey+F=0
for various values of the coefficients, and then proceed to three dimensional shapes. This exercise would take too long by hand and most students do not have the draughting skills to cope. With a computer the principles can be investigated without busy-work sketching exercises.

Of course, students should be able to do hand calculations and hand sketches. But, when a series of calculations and sketches is assigned, the first few exercises can be designated hand exercises (to establish that students have acquired hand skills), while the rest of the exercises (those designed to encourage mastery of principles, discovery, a sense of experimentation and wonder) will be designated calculator and computer exercises. The mechanics of such exercises would necessitate the availability of programs and machines. But the software and the machines will not become available until mathematics teachers show themselves able, ready, and eager to use them. Mathematics teachers must make it generally known that such materials are an absolutely essential part of their equipment ...not for the sake of technological training (that is the duty of educational computer specialists), but for the sake of adequate, modern mathematics instruction ...for the job which they are hired to perform.

FAST, FAST, FAST RELIEF

Although most mathematics teachers encourage their students to use calculators in algebra, only a very few actually give lessons on their use and only 40% agree that the use of the calculator should be emphasized (College Algebra Topic Specific Questionnaire).

Sixty percent allow students to use calculators at all levels of work in trigonometry (exercises, homework, exams) but very few seem to demand the use of calculators (Trigonometry Topic Specific Questionnaire).

Seventy-seven percent of teachers used calculators in Analytic Geometry (59% in ONTARIO 12 to 82% in USA PRECALCULUS) but that seems to be as far as it went (Analytic Geometry Topic Specific Questionnaire).

This is the story on calculators, which have been around for years. What hope is there for computers when teachers seem so under-whelmed by the old technology.

The General Classroom Practices Questionnaire gives us some more clues as to the source of the problem. Eighty percent of the teachers reported that it would be difficult for them to do without text books. Almost 80% of the teachers reported that it would be difficult to do without the recollection of the mathematics courses they studied when they were students. Only 20% felt it would be difficult to do without professional journals and books not used by the students and fewer than 10% felt it would be difficult to do without professional meetings, inservice or workshops.

IT FIGURES!

In other words, many mathematics teachers believe they were prepared once and for all time by mathematics courses taken 5, 15, and 25 years ago. Mathematicians don't believe this. Businessmen don't! Academics don't! We are going to have to utilize the computer revolution simply to be able to do enough to compete. Yet, the closest many teachers have come to upgrading themselves is to suffer through a change in text books and courses of studies. Many people put their careers on the line every work day - by performing new surgical procedures, by utilizing new information and machines. Can we do less? Re-qualifying exams for Mathematics teachers would give us fast, fast, fast relief from the anachronisms of 'schoolmath' and at the same time alert the public and our students that we are serious when we point to a machine and say, IT FIGURES, I THINK!

(Graphs, cartoons, and software demonstrations accompany the presentation.)

A Model For Inservice Training of Teachers That Effects Real Change

Jim Burns
The Dufferin-Peel Roman Catholic Separate School Board
1150 Eglinton Avenue East
Mississauga, Ontario L4W 2M6

An Inservice Training Laboratory (ITL) is used to train teachers in the use of the computer in the classroom. This method is unique in that it is done at a school over a month long period with a full time consultant, detailed, curriculum, software, and 16 networked computers.

The development of growth on the part of an individual or an institution should always be a goal. This growth, whether it be minor behaviour modification sought by an individual or a major shift in life or career focus involves the process of change. This is also reflected in institutions whether it is a minor marketing shift for a standard product or a major advertising campaign to launch a completely new product.

The phrase 'change is a process not an event' (Fullan and Park, 1981) highlights an aspect to this process of growth that is poorly understood by individuals and institutions not directly influenced by the market place.

Education in recent years along with other institutions and professions has been forced to become more accountable. This has evolved ironically in part through the success of such institutions as education. The average adult is now better educated and hence more aware of their rights. The other part bears on the consumer aspect in that he/she is now more aware of the market place and its built in accountability. This has been facilitated greatly by the advent of the new communications technology. Thus, with a rise in education and information level, expectations have fostered a legitimate emphasis on accountability.

The reaction in part in the education field has been more courses for teachers, more clarity around expectations such as Bill 82, and an attempt to be more responsive to consumers. This different focus for such an institution as education requires an understanding of the change process. As educators, we have excelled in development work such as curriculum. We have addressed the area of accountability by placing an emphasis on objectives. The next stage was a refining of the objectives to the point where there were measurable outcomes.

Once outcome measures were more easily related to curriculum objectives a major gap was identified. There developed questions of analysis as to the disparity between objectives and outcomes. Was there a misconception of the intended objective? Were the entry characteristics of the client realistically reflected in the objective? Was the impact of teacher style such as facilitator or instructor, accounted for in both objectives and outcome measures?

Education has addressed development in various ways with objectives being one of them. The institution in responding to the consumer has also developed sophisticated review procedures with testing instruments and sampling methods. There remains a gap in the change process.

This gap between the development of a product, whether it be curriculum in education or a new consumer electronic device in the market place, and the review of its market success has long been identified in business and industry as implementation. It seems ludicrous to have to say it but for clarity of the paper it must be said, that a person or institution must do more than develop a desired end product and a review procedure to measure its success as related to the goal objectives. The intermediate step of implementation must be identified and planned for in the whole process of change. Business and industry have learned this and hence identify a large percentage of product development costs for market research, advertising, and consumer training if necessary.

Although one could continue this comparison of parallel process development in fields such as business and industry as opposed to service institutions which are one step removed from the market place because of the profit motive, the length of this paper does not permit such discussion. The intent is to present an implementation model developed in The Dufferin-Peel Roman Catholic Separate School Board of Education for the inservice training of teachers in the use of computers in the classroom. This Board of approximately 40 000 students and 2000 teachers is located just west of Toronto, Ontario. The present computer ratio is approximately 1 computer for every 50 students with the majority being in the secondary panel. The elementary schools each have a minimum of one computer and range all the way up to 16.

The initial phase of the long range plan of integration of the use of the new technology into the classroom was identified as teacher inservice training for elementary school teachers. Computer literacy curriculum followed by software and hardware were the other phases. Each of these was placed on separate tracks. The order is not a question of sequence but more of emphasis. All phases are at various levels of priority at any one time. Initially as mentioned the inservice training of teachers was a priority.

A word should also be mentioned on the integration of the computer into the curriculum. This, of course, is the long term goal but realistically certain thresholds in terms of hardware and software must be met before integration becomes an achievable objective. Stages that precede integration are the threshold of reality and literacy. The reality threshold simply refers to a basic amount of equipment in each school so that everyone including students and teachers can take the new technology seriously. The literacy threshold requires more hardware to enable full class participation in an almost unscheduled environment. Finally, with enough hardware and software, the integration threshold can be considered. The stages are sequential and are chiefly demarcated by the amount of software and hardware.

To return to the topic at hand, teachers inservice training has been addressed in a plethora of arenas. There are univeristy, community college, television, Ministry of Education, and Boards of Education courses. The order given also identifies which courses best address the needs of the teacher,going from the theoretical university course to the tailored Board of Education workshop. Too often, however, teachers have developed an enthusiasm for the use of computers through all of these courses but have to face the reality of very little hardware that is both user friendly and software 'rich'.

The approach The Dufferin-Peel Roman Catholic Separate School Board took was to design a model that would provide the necessary content information, hardware, and quality software. Other aspects were also built into the model to facilitate the change process such as the most effective population unit to work with. It was found through research (Bruce Joyce) that the school staff unit was the best. In addition, the ease of taking the course was addressed by providing it at the school in such a way as to permit teachers who took it on their own time without extra pay to be home for supper.

The course consists of the following sessions;

1. Introduction
   This model of inservice training is explained along with a demonstration by participants of the capabilities of the ICON.

2. Education Applications
   Participants learn to categorize types of educational software through hands-on use of the ICON.

3. Word processing
   The edit program is explained.

4. Problem-solving/Programming
   Artificial intelligence and logo are introduced.

5. Graphics/Modification of a Program/Curriculum Integration
   A graphics package is demonstrated and used by participants. Also they are shown how to modify a program for their class.

6. Evaluation of Software/Computer Literacy Curriculum
   The evaluation of one piece of educational software is used to introduce a checklist. Also the Board developed Computer Literacy Curriculum is introduced.

7. Instructional Strategies/Lesson Planning
   Various instructional strategies with the computer are introduced.

8. Computers and the Curriculum
   Integration of the computer into the curriculum and individual lesson is discussed.

The most crucial part of the model, however, was the opportunity for the teacher to take what he/she learned in the inservice training course and teach his/her own class with a full time coach available. Thus, the teachers took the 16 hour course, 2 hours a night for 2 nights a week for 4 weeks, and then taught their own class two modules with the consultant in attendance to provide assistance if any hardware or software problems arose. A module is defined as three one hour sessions on a particular topic. Different manuals were provided depending on whether the teachers taught primary JK - 3, junior 4 - 6, or intermediate 7 - 9 classes. Each manual contains the inservice training handouts, manuals for the major pieces of software, detailed lesson plans and all the activity cards required.

The model is called an Inservice Training Laboratory (ITL). It was piloted from January to June of 1984 in five schools. It includes 16 computers and all related equipment, even tables and posters.

This model provided the mechanisms for change on the part of the teacher, that is, the development of a comfort level and certain skills. It could only accomplish this through a minimum of a month long visit of all the laboratory equipment and a full time consultant during the day. Another consultant would provide the after-school instruction for the 16 hour course. The change process requires time because teachers have to recognize the need for change and then work through the process for themselves with the coach as facilitator.

The full program of this model began in November 1984 with six schools going through the process simultaneously. This is a two year program to enable all schools to be serviced which requires a large commitment in terms of resources. The equipment which is the Ontario Approved Educational Microcomputer - ICON is less costly in comparison to the personnel commitment of two consultative staff for each laboratory.

The original intention was to leave equipment behind in each school once the month long immersion had taken place. For many reasons there is a time lapse between the end of the course and delivery of the equipment. This will have to be addressed by further inservice training once the equipment is sent to the school.

The pilot and full program phases both highlighted major side benefits to this model. The school staff experienced a renewal as a unit as they worked together over this extended period to come to an understanding of the place of the new technology in the classroom. Teachers gradually through the month long change process gained a confidence level to the extent that they saw themselves as forces for change or change agents in relation to the use of the computer instead of bystanders. This was measured by the evaluations, demands on present in-school computer resources, and module selection. As mentioned, teachers taught two modules to their class during the period the ITL was at their school. At the end of whichever of the prepared modules the teachers chose first they were given the option to extend that module for a further three lessons or pick one of the other prepared modules. The consultants would assist them with this extension. The goal of comfort level was met as the majority of teachers chose the former.

Another important outgrowth to the model was the involvement of the community. Community and parent sessions were conducted by teachers who were newly trained through the model. This outreach became a very positive force which reflected back on the school staff unit.

Thus the development of growth requires a slow process of change. As part of the change the implementation model must provide content, an opportunity to simulate or use what is learned, a coach for security, and appropriate feedback such as student enthusiasm.

In this way the long range goal of the integrated use of computers in the classroom has many phases; inservice training of staff, curriculum from literacy to integration, software evaluation and development, and hardware selection and acquisition. At any given point, as each phase moves ahead on its own track one will have a priority over the others. To us the natural place to start was with training of teachers. The effectiveness of the ITL has resulted from an awareness from the outset of the importance of the change process components of content, simulation, coaching and feedback.

Fullan, M., Park, P. Curriculum Implementation, 1981, Ministry of Education Ontario.

Joyce, Bruce, Showers, B., "Improving Inservice Training", Flexibility in Teaching - An Excursion into the Nature of Teaching and Training, Longman.

An Assessment Of CAI In The Field Of ESL

Raoul J. Freeman
California State University - Dominquez Hills
1000 East Victoria Street
Carson, CA 90747

An analysis is made of existing CAI systems which can be used to teach English as a second language to Hispanic students. Although various systems offer certain advantages it seems that there is need for further development. The attributes of what is required are described. These include direct linkage to instructional objectives, use of a notional syllabus, use of a programming language that is not hardware specific, an effective instructional management system, use of high quality graphics and integrated audio, etc. The technology for such a development exists, and it is recommended that such a venture be pursued.

This paper presents the findings of a study to ascertain:

1. whether any existing computer assisted instruction (CAI) system could be effectively used to enhance the teaching of "English as a Second Language" (ESL) to Hispanic students.
2. the attributes of a desirable CAI system for teaching ESL to a Hispanic student population.
3. the feasibility of undertaking the development of a desirable system.

The effort to survey the existing state of the art included visits and/or contacts with various computer hardware manufacturers, software developers, publishers, educational research laboratories, software collection and dissemination agencies, school districts, professional societies, and colleagues across the nation. In addition, various software directories were researched, and computer access to national on-line databases of software listings was established

OFFERINGS VERSUS REQUIREMENTS

Analysis of current computer based activity indicates that the potential for a computer revolution of education exists. Prices of computer hardware have been dramatically reduced, teacher awareness and willingness to use computers has increased, and the advent of microcomputers has caused a bandwagon effect for use of computers in education. The biggest stumbling block to widespread implementation of effective CAI is the lack of comprehensive software that is integrated to learning objectives, covers vast areas of curriculum, functions as a cohesive learning system, runs on most popular brands of computer hardware, and is inexpensive. It is possible now to dispose of that stumbling block and to reap the large potential which computers can bring to our country's educational program.

The current situation in the United States is that there is educational CAI software available. However, most of it is relatively disjoint, concentrates on specific topics, lacks integration with other software in terms of objectives, reinforcement, record keeping, and conventions, and has limited applicability between different makes of computer hardware. There are a few integrated systems available, and they do offer certain advantages. A comparison of such systems is given in Electronic Learning (1).

A survey of existing material resulted in the finding that the available amount of CAI software for ESL for

Hispanic students was rather meager. For example, less than 0.1 percent of all the programs listed in the RICE national data base of educational software were related to ESL. A comprehensive listing of CAI software of ESL for Hispanic students is given by Sauve and Schnuer (2). Except for the CCC and CDC offerings which are discussed below, there is usually no instructional management system included. Both the integrated and non-integrated software have in many cases not made explicit the educational objectives for which they are designed. When objectives are stated, there is no explicit, pre-determined correspondence to the published instructional objectives of given Districts, for example such as those of Los Angeles (3). The only two integrated systems that are available to teach ESL to Hispanic students are those of CDC-PLATO (4) and CCC (5). Of these two offerings, the CDC one offers graphics, while the CCC one does not. Each of these systems is limited to a specific brand of hardware (coincidentally that of the sponsoring corporate entity). The CDC audio is more rudimentary than that of CCC. Both offerings provide excellent instructional management software. The CDC system makes a first try at a notional syllabus, whereas the CCC system uses a conventional subject syllabus approach. It is questionable whether either system is geared to students at the lowest grade levels. Both systems were designed to teach ESL to Hispanic students, but neither can lay claim to having been custom-made to meet existing teaching objectives of client Districts. However, this is not meant to say that these systems necessarily do not cover the objectives of a given District.

What is required to overcome the current weaknesses is a new system which will be linked directly to teaching objectives, written in a language that is not hardware specific, making use of a notional syllabus, and having applications in real-life settings. The system must have an effective management or record keeping and reporting capability, be based on strands and sequences with appropriate reinforcement criteria, make extensive use of high quality graphics and integrated audio, and provide reliability and ease of use for students. Furthermore, such a system must have comprehensive and effective user support materials, accurate content, educational value, well defined purpose, clear and logical presentation, freedom from race, ethnic, sex, and other stereotypes, a level of difficulty appropriate to target grade levels, proper feedback on student response, capability to stimulate student creativity, and capacity to allow the learner to control the rate and sequence of presentation and review. The software system that is developed should be capable of running on a shared hard disk configuration with dispersed terminals, as well as in a computer laboratory setting. The hardware must support the audio, graphics, hard copy and other requirements. Furthermore, an adequate authoring system must be provided to facilitate writing of the software.

DESIRABLE SYSTEM CHARACTERISTICS

An outstanding CAI System for ESL still remains to be developed. The essential elements of such a system would include direct linkage to instructional objectives, usability among various brands of hardware, notional syllabus, comprehensive grammar and vocabulary coverage, educationally sound application software, a management system, an authoring language, and high quality graphics and audio.

Linkage to Teaching Objectives

One of the foundation requirements for the new system is the direct linkage of the computerized teaching process to instructional objective. This would make the software relate to specific instructional objectives (i.e. skills that are to be taught and tested against). This requires the translation of presently existing instructional objectives into machine processable form. A method to do this is described by Freeman (7).

Usability on Various Computers

In order to assure the development on a basis that can be broadly used, the software should be written so that it is transportable across various brands of hardware. One way to accomplish this is to develop the applications software in the "p" language. This makes it transportable across most popular types of microcomputers including IBM, Apple, DEC, TI, Commodore, TRS-80, etc. Thus the development would be such that it will be usable by Districts across the country regardless of what brand of computer they have previously acquired or are going to opt for in the future. A more in-depth explanation of the "p" language is given by Buczek (8).

The system is based on the concept of a pseudocode interpreter. Programming is compiled, not into the machine code of a particular CPU, but into a standard optimized code, called a pseudo machine code. This code does not execute directly on any standard processor but is executed by running a p-machine emulator. The latter is a machine-language interpreter that must be written for each different CPU on which the system will run. Writing this p-machine emulator typically requires only a small fraction of the effort that would be required to create an entirely new compiler for each machine, and such emulators are already available for most major brands of hardware that would be of interest.

Notional Syllabus

Another fundamental departure that the new system will take from more classical offerings is the use of a notional syllabus in the teaching of language. Prime uses of language relate to requesting and giving information, expressing thought processes and opinions, making judgments, modifying people's behavior, expressing personal feelings, and interacting socially. Vocabulary and grammar are viewed as a means to accomplish language uses or functions. An early delineation of the concept of notional syllabus was developed by D.A. Wilkins (6). The topics to be taught then are the notions or functions rather than the mechanics of vocabulary and grammar as ends in themselves.

Vocabulary and Grammar Coverage

In the context of the ideal ESL course, it is especially important to choose that vocabulary which is the key to the language functions that are being taught. Grammar topics need to include punctuation mechanics, types and usage of classes of words (e.g., nouns, verbs, adjectives, pronouns, etc.), tenses of verbs, clauses, questions, etc. The applications or settings in which the grammar and vocabulary is to be applied must be made explicit. In order to give full meaning to a language, it is important to cover a variety of settings or themes (e.g., the family, the city, the office, the world, the store, etc.). Thus the themes would take the teaching objectives, which include functions, across various areas such as the home, school, etc. It is only by such illustrations that the full richness of the subject matter inherent in the teaching objectives can be made available to the student.

Audio and Graphics

Use of audio in the teaching of ESL is essential. Audio to digital conversion and concomitant adequate storage capacity is required. For an adequate ESL course, even using compressed data, it is easy to visualize a requirement for 30 to 50 megs of disk storage for the audio alone. Having a shared disk local network would amortize the cost of this storage over multiple users. Other equipment necessary would include a set of earphones at each learning station. The possibilities opened by computerized voice input and output are staggering. They range from teaching ESL far better than is currently possible, all the way to tutorial computerized instruction of pronunciation. Graphics must be utilized in order to make the material interesting for the students. Continuous text just creates boredom, whereas animation maintains interest.

Management System

The software will be organized into strands and sequences. A management or tracking system will keep track of where every student is, the results achieved, etc. From this base of data, the computer will be able to assign appropriate work to the student in future computer sessions. The system will be capable of issuing various reports. For example, the type of data kept will include the number of sessions the student has taken in the course, cumulative time spent, number of time-outs, percentage of exercises answered correctly, grade placement, test scores, percent of work mastered in instructional strands as a function of time on a cumulative basis, etc.

Based on this raw data, various class and school reports by teacher, by grade level, student sex, etc. will be available. Also reports to parents can be made from such a data bank. This repository of data also would be used by the system to develop learning prescriptions for students. Reports will be able to be requested by teachers, administrators, etc., on demand.

Authoring Language

To facilitate the writing of the application software it is essential that an authoring language be available. Such a language permits English-like interaction between the computer and the author of the courseware, thereby easing the burden of programming significantly.

RECOMMENDATION

Grant and Eiden (9) indicate that there are close to five million Hispanic- origin ESL students in the United States. Computer assisted instruction in the ESL area could fill a large gap in the educational process for these Hispanic youngsters. All relevant technological ingredients required to put together exemplary CAI courseware for ESL are currently available. It is recommended that the development of appropriate computer courseware for ESL be undertaken.

ACKNOWLEDGEMENT

The past support of the Weingart Foundation for the work of the author is gratefully acknowledged.

REFERENCES

1. ----------, Electronic Learning, 2:5, pp. 72-76, 1983.
2. D.D. Sauve and S. Schnuer, Guide to Microcomputer Courseware for Bilingual Education and Related Settings, National Clearinghouse for Bilingual Education, Rosslyn, Virginia, 1983.
3. -----------, Elementary School Curriculum: a Balanced Program, Los Angeles Unified School District, Los Angeles, California, 1979.
4. -----------, ESL Curriculum-PLATO, Control Data Corporation, Minneapolis, Minnesota, 1983.
5. ----------, Teacher's Handbook for English as a Second Language, Computer Curriculum Corp., Palo Alto, California, 1983
6. D.D. Wilkins, Notional Syllabuses: A Taxonomy and its Relevance to Foreign Language Curriculum Development, Oxford University Press, London, England, 1976.
7. R.J. Freeman, Definition and Evaluation of Computer Support Systems for Instruction, AEDS Journal, 17:4, pp.46-55, 1984.
8. J.R. Buczek, The Power, Incider, 1:1, pp. 114-126, 1983.
9. V.W. Grant and L.J. Eiden, Digest of Education Statistics-1983, National Center for Education Statistics, Washington, D.C., 1982.

Incorporating Computer Simulation Methods Into Classroom Instruction: From Social Studies And Mathematics To Emergency Medicine

Dr. Brent E. Wholeben
University of Texas at El Paso
Department of Educational Leadership and Counselling
The University of Texas at El Paso
El Paso, Texas 79968

Simulation is one of eight computer environments which facilitate the instructional process. Instructional methods based upon computer simulated 'learning' have proven very effective for 'reality based learning' in mathematics, the social sciences, engineering, business, medicine, and the laboratory sciences. With computer based simulation, instruction becomes a complex interaction between (1) decisions based upon information, (2) informational analysis based upon the application of multivariate principles, (3) continuous modification of the learning environment based upon decisioning trends, and (4) immediate feedback to the learner for formulating subsequent decisions.

INTRODUCTION

Although simulation software has the potential to address either or both of the needs of application and instruction, the use of such software during field based applications will be different than its deployment as an instructional tool, exhibiting different attributes, and therefore requiring a different perspective on the part of an evaluator whose goal is to assess the relative strengths and weaknesses associated with the varying 'use potentials' of such courseware. As an instructional tool, simulation courseware must be valid as a teaching medium in addition to its validity and reliability for empirically solving a problem situation.

COMPUTERIZED SIMULATION

Simulation, as a technical process, supports the philosophy that problem resolution is a complex activity requiring a highly structured approach for understanding (1) systemic interrelationships based upon subsystemic influences, (2) what alternative solutions are viable, (3) the likely impact of each of these alternatives upon the problem defined, and (4) transactional effects within the system as a whole based upon the alternative selected. To simulate means 'to test' alternative actions based upon what one knows about the problem, the system within which the problem resides, the potential significance of alternate actions in resolving this problem, and other related effects upon the system which such solutions might effect during implementation. Simulation allows the teacher-user to employ basic assumptions about the environment of the problem, and the influence(s) of alternate solutions upon that problem as well as the environment within which the problem resides. From such a comprehensive understanding, likely effects of alternative decisions, i.e. alternate solutions, can be tested for their validity and reliability prior to actual empirical implementation.

The structuring of the problem-resident environment for decisioning simulation requires the construction of a model, typically of quantitative dimensions, allowing the planner to enter data regarding (1) the environment, (2) the context of the problem itself, and finally (3) intended (alternate) solution(s), and then receive information which suggests the likely impact which such an intended solution would have upon the problem in particular, and the environment within the problem resides in general.

Data requirements are of particular interest to the teacher-user when employing computer simulation techniques. Certain data requirements which facilitate simulation include:

01 establish conditions (i.e. interactions) pertinent to the decisional environment, and parameters (i.e. limits) germane to these environment interactions;

02 modify existing conditions and/or parameters based upon changing needs or predictive trends existent within the decisional setting;

03 query the impact of alternate activities (actions) upon the environment, including the identified problem, and based upon the particular alternative action being simulated and its effects upon associated criterion based indices;

04 assess the sensitivity of current solution(s) with respect to projected variability (changes) in established conditions and/or parameters for the future; and,

05 evaluate the significance, statistically and magnitudinally, of the results of a simulated decision for both adequacy of impact as well as appropriateness of effect.

The ability of a computer based simulation technique to achieve the above stated objectives will largely depend upon the capabilities of the software available to the user.

GOALS OF SIMULATION COURSEWARE

While simulation software will normally function as instructional courseware in the classroom as well as applications software in the field, instructional simulation courseware must typically be more narrowly defined. In addition, some simulation based courseware, if in fact they could be used to approach real world (field based) problem settings, are usually limited because of such technical problems as the number, type and range of parameters and/or conditions necessary to model the problem environment adequately and appropriately.

Most computer software which performs simulation-related analyses for field-based applications can be used in the classroom for instructional purposes, both in terms of demonstrating the utility of such an approach as well as training the student how to use that particular software package. On the other hand, more instructionally oriented software packages often can be utilized for field based applications, although many such packages are very limited in the size of problems they might effectively address.

As an instructional intervention tool, computer simulation courseware is often concerned with common goals:

01 demonstrate relationships which exist within structured systems based upon interventions introduced within that environment, and the effect of those interventions;

02 practice the technique of structured intervention within a strictly defined environment, observing the variable impact based upon varying focused interventions;

03 review the results of well-defined interventions within a highly structured, transactional system, and the foundations for the validity and reliability of these results;

04 study the effects of alternative actions, based upon certain criterion indices which signal strengths as well as weaknesses associated with each alternate tactic; and,

05 experience the utility and style of distinct, controlled reactions based upon a structured intervention, and the predictable reliabi-

lity of criterion based indices which assess the importance and/or significance of these reactions.

STANDARDS FOR SIMULATION COURSEWARE

Simulation-based courseware utilizes the computerized environment to interact with a student in a reality-based orientation. Simply stated, the student makes decisions and enters appropriate data relative to these decisions, in response to which the computer analyzes the probable impact of such decisions in a controlled though quasi real-life situation, and then provides feedback to the student as to what impact such a decision would have 'really' had in real-life milieu.

PRESENT A CONCISE, CLEAR SITUATIONAL CONTEXT FOR THE SIMULATION WHICH IS LEARNING RATHER THAN GAMING BASED;

Although instructional simulations can be intrinsically entertaining, especially those very skewed in terms of their mission of teaching priority, the amount of student learning achieved is the final indicator of efficacy. The context of the 'situation modeled' must be clear at the beginning, so that the user understands the environment in which decisions will be made, and thereby better understands the reactions of the system based upon these decisions. Such demands will typically require that a student's responses be cumulative, that is, once a decision is performed, subsequent responses from the system to the student include the results of these precedent judgements.

PROVIDE A CLEAR INDICATION OF VARIABLES AND CONSTANTS, THE EXTENT TO WHICH THESE VARIABLES ARE USER VERSUS PROGRAM CONTROLLED, AND THE RELATIONSHIP BETWEEN THESE TWO TYPES OF VARIABLES;

Although empirical reality is seldom as predictable as planners might like, instructionally based simulations must be predictable in terms of their responses and/or reactions to student decisional inputs, if the student is to learn and prosper from such reactions. Student understanding is facilitated through as clear an understanding as possible, initially, of those variables which the simulation entertains as analytical apart from those measures which are always in effect as constants. Subsequently, the student must understand which of the variables can be directly controlled (i.e. measures input) by the student from the keyboard, and which measures will be generated by the simulation in response to student input via other variables. Finally, student knowledge concerning how such variables relate to each other fosters a learning environment which increasingly makes subsequent reactions of the environment predictable. During subsequent instructional simulation episodes, the student can be given less information, progressively, and encouraged to developed personal perceptions related to the interrelationships existent within the problem environment under study.

PROMOTE AN ACCEPTABLE LEVEL OF PERSONAL INTEREST ON THE PART OF THE STUDENT USER VIA THE VIDEO DISPLAY SCREEN FORMAT, ESPECIALLY IN TERMS OF COMPREHENSIVENESS, EFFECTIVENESS AND EFFICIENCY OF INFORMATION DISPLAY;

Similar to other forms of 'physical' documentation, the screen display should be clear and concise, making available whatever information the student needs in order to refine previous judgements and/or modify subsequent data input demands of the system. Although the student should be required to make specific demands of the systems for specific information, such information should be easily interpreted to facilitate the student's continued use of the decisional simulation.

ALLOW AN ON-DEMAND LINKAGE TO TUTORIAL MODE, PROVIDING ADDITIONAL INFORMATION WHY CERTAIN EFFECTS CAN BE EXPECTED BASED UPON A STUDENT'S CHOICE OF ACTION OR INACTION, AND FURTHERMORE WHAT MIGHT HAVE BEEN A WISER RESPONSE BASED UPON CURRENTLY RECORDED CONDITIONS;

Simulation courseware must be first and

foremost didactic in nature, and therefore promote an environment in which the student learns from simulated experiences. When the student, for example, feels that any further judgements concerning data input would be little more than mere guessing, the student should have the ability to ask the system (1) "What might I do next?", and (2) "Why?". In addition, if the system responds to a decision with a negative reaction, and the student fails to understand the validity of this reaction, means should be available to query the system for an extended explanation of rationale.

GRADUATE THE PROGRAM'S SITUATIONAL COMPLEXITY AND RESPONSE ATTRIBUTES BASED UPON A 'NOVICE ... EXPERT' CONTINUUM, ALLOWING BEGINNING USERS MORE STRUCTURE DURING THE INITIAL LEARNING PROCESS WHILE PROVIDING SEASONED USERS WITH MORE COMPLICATED SIMULATION PROBLEMS;

Since computerized processes provide the potential for individual students to proceed individually through the teaching-learning paradigm, the ability of instructional simulation programs to 'start simple' and become progressively more difficult based upon a student's individual performance trends is considered important.

VARY THE TYPES OF RESPONSEFRAMEWORKS (USER RESPONSE OR QUERIES) TO BETTER APPROXIMATE REAL-LIFE SITUATIONS (E.G. KEYWORD, DICHOTOMOUS, DIRECTIONAL, FORCED CHOICED AND SHORT-RESPONSE);

The key to experiential learning through computerized decision-modeling simulations can be attributed to the need of the student to interpret new information output as it transpires -- that is, respond to new information based upon a progressive understanding of the system under simulation. By varying how such information is presented to the user, and how the user must respond to differing situations, simulation provides a realistic context to problem solving which approximates real life situations.

COORDINATE A STANDARDIZATION OF PROCESS (DISPLAY, RESPONSE, ETC.) WHICH ALLOWS THE STUDENT TO LEARN CAUSE-EFFECT RELATIONSHIPS WITHOUT UNDUELY COMPLICATING THE INTERACTION WITH SEEMINGLY AT-RANDOM OCCURRENCES.

Corollary to the previous criterion, a variance in response or reaction frameworks, and the student response to that framework, should not become so random as to defeat the initial priority mission of the simulation, that of instruction and student learning. Without a standard structure to the computer simulation, many students would be unable to interpret the underlying concepts and theories presented for learning consumption.

FUTURE OF INSTRUCTIONAL SIMULATIONS

The seeming complexity inherent in simulation modeling, the lack of understanding on the part of teachers, the reluctance to embrace any approach which is quantitatively oriented, and the general fear of computers themselves -- cumulatively militate against the acceptance of simulations as viable teaching tools. Recently however, such reluctance has generally eroded through the increasing acceptance of the computer as an effective and efficient information processing device.

- NOTE -

For additional information on the use of computer simulations for teaching and administrative decisionmaking, contact the author directly:

Brent E. Wholeben, Ph.D., E.M.T.
Department of Educational Leadership
University of Texas at El Paso
El Paso, Texas 79968
(915) 747-5300/5355

Local Area Networks:
Present and Future Possibilities for Classroom Use

Jo Ann Wilton
Ontario Institute for Studies in Education
and Peel Board of Education
Mississauga, Ontario

With increasing frequency, both the traditional and the new generation of computers are being organized into local area networks within schools or within classrooms. Although the technology exists to assemble a variety of configurations, use of such networks will be limited to sharing disk storage and printers unless appropriate software for other applications is developed. It is useful to review the possible roles for local area networks in classrooms and focus on the implications for software design and curricular use.

Networks in various forms have been around for quite a while, but several factors are combining to make classroom networks more common and more powerful. Several exciting new applications of computers are made possible and traditional ones enhanced through networks. These impacts can be subdivided according to three particular characteristics of networked environments which make them possible: increased storage space, more flexible operating system and node to node communications.

What are computer networks?

Soon after Computer Studies first appeared in the secondary school curriculum, a few Boards of Education moved from remote processing of mark-sense cards to time-sharing networks with in-house terminals accessing off-site mainframes or minis for batch processing of students' work. But wider computer use in schools and the resulting meteoric rise in the numbers of students electing Computer Studies courses did not really begin until the late 1970's when microcomputers became widely and inexpensively available.

The flexibility and portability of the stand-alone machines brought the power and mystique of the computer world to the individual user and that self-standing capability was praised and protected. Teachers and students revelled in the real-time, interactive, graphic mode of programming offered by microcomputers and only occasionally grumbled about the small capacity or the slow speed of tape or early disk drives. Not everyone was completely sold on the new environment, though, and many teachers continued to use terminals as part of a mainframe network.

Outside computer studies classrooms, some of the earliest uses of computers in instruction involved large CAI packages delivered through networks from a mini or mainframe in a remote location to user terminals in classrooms. The cost of this mode worked against its wider adoption particularly at a time when school boards were giving priority to studying about computers rather than studying with computers. Now the whole nature of computer networks has and is changing coincident with changes in classroom expectations for computer performance.

When used in the popular literature, the term networking can vary in the way it is used. It is frequently used as a non-technical reference to intercomputer communication such as electronic mail or data base access ; yet, is also used to refer to a quite simple technical linking of stand-alone processors with shared peripherals such as disk-drives and printers.

A discussion of network differences is not the purpose of this paper; but, has been attempted separately by the author in doctoral thesis work. As a basis for this discussion of local area networks in education, a set of working definitions is offered. Local area network (L.A.N.) shall refer to networks extending over small areas (less than 1000m in diameter) in which each computing unit has independent computing power and capacity and has access to all nodes. Nodes in this case could be computing units or support peripherals such as disk storage or printers or a gateway out of the network such as a modem. The communication to all nodes may be either direct or facilitated through files held on a central storage unit. The former situation, referred to here as a fully integrated L.A.N., is ideal and increasingly available to classrooms; but, the latter (a disk-sharing L.A.N.) is much more common in today's classrooms.

Disk-sharing networks, requiring less sophisticated linkages and limited operating software require less powerful computing units and are relatively easy to set up with most existing computers. Different commercial networking packages offer variations in terms of what types of linkages are possible and what types of protocol are used to allow communications. Where networks have been rejected for classroom use, the rationale has generally hinged on one of three factors: i) a perception of greater cost for networks, ii) a philosophical desire to maintain the independence and individual integrity of each user station, or iii) dissatisfaction with the limitations on free communications which slow some network systems or, more importantly, even prevent the use of specific commercial software on the network.

On the other hand, fully integrated local area networks can only be established where the operating system which links the shared storage and the computing units is designed to treat all nodes as equal. Currently, the ICON network based on QNX and an IBM network using PORT as the operating system are available to schools - the latter only in a test mode at this time. The ICON system was designed as a fully integrated network system from the start; whereas, the PORT system is an add-on to a configuration of IBM computing units centred by a PC-XT. The trend seems likely to continue with more sophisticated models of microcomputers incorporating local area network capability as an integral part of the original design

Three conditions contribute to the growth of networks in education at the present time:

1. With the increased number of user stations required in each school, some schools are returning to the time-shared terminal strategy with a micro, mini or mainframe central processor.
2. The traditional linking methods of floppy disk sharing are spreading beyond the computer studies classrooms to uses across the curriculum including many elementary environments.
3. A new generation of microcomputers with greater capabilities and their own network structures is being introduced to the school systems.

Layers of networks for education

Within the definition of a local area network, computers can be linked with the entire system *inside one classroom* as is usually the case with disk-sharing systems or *distributed throughout a school.* In this second case, the major storage unit would be in a central location but user stations and other peripherals including printers could be placed in classrooms in any combinations deemed to be appropriate to that school.

Local area networks can be linked into distributed networks for sharing resources, either computer or curriculum resources, *among schools or throughout a school board.* In Ontario there is a *province-wide educational network,* ECNO (Educational Computing Network of Ontario), into which it is anticipated schools or school boards can link their local area networks for exchange of information, access to data bases or distribution of software packages.

Using networks in education

In the early thinking about microcomputers, it was expected that all of the necessary peripherals and other capabilities would be right at and unique to each user station, but networks even in the most simple form offer an alternative facility for providing that level of support.. If computer networks are used for no other reason in our schools over the next three to five years, the rationale of extending user resources will still ensure that networks multiply and become more reliable and user friendly.

However, the additional possibilities which networking offers can add new dimensions to the use of computers in the curriculum. Each of these developments presents further implications in terms of software, teacher training and computer support structures.

1. Impact of Increased Storage Space

a) CAI/CMI Revival

Access to larger disk storage in a network configuration opens the possibility of courseware (traditional C.A.I.) or unitware being used on a wider basis in classrooms. Some CAI software does exist now but not on the types of computers in use in schools. However, the Ministry of Education for Ontario is funding projects to develop such materials both through the Computers in Education Centre software development processes and simultaneously through the Correspondence Division for distributed education uses.

Even if products on the scale of courseware do not become available, increased storage areas make possible, even easy, the software management of student use with record keeping, tracking of difficulties and readjustment of individual learning paces and routes. With the probable rejuvenation of CAI and/or CMI, concern should be given to blending what has been learned about computers and children through extensive lessonware use with the experience and design protocols of traditional CAI work.

b) Curriculum Oriented Data Bases

Access to data bases is gaining in strength as a major benefit of computer use in schools. This focus parallels the rise in importance of this dimension of computer technology in business and government circles. The concepts of data base structure and related skills can be developed through small data base packages on stand-alone computers. Several good examples are in use in Ontario schools now. However, except in the case of very specialized information, data bases to be searched for their content and used as a resource by the students will be too big for individual microcomputers or the floppy disk drives used in most schools now. Outside data bases on the scale of InfoGlobe and Dialogue can be accessed through a modem by any microcomputer; but, data bases of use in a specific unit or a course, either acquired as a resource package or built by the students , can be held on a hard disk of sufficient capacity as an integral part of an in-school network.

2. Operating System Structure

a) In-school Software Management

When software was scarce and good educational software was even less available, tapes or disks could be handled in the teacher's desk or on a shelf. Load time was short because programs were short. Ironically, many good programs were discounted because they took too long to load. Schools will now have to develop an in-house process for managing the growing library software - where to store it, how to access it and how to let people know about it. With a network, only one copy needs to be in use and the disk menu or inventory can be structured to suit the school, course or class.

b) System Based Software

When a microcomputer needed every little bit of memory space it could maintain, efficiency of programming and use of operating space were paramount. Now, system 'tools' such as editing systems, graphics design and data file managers can be available immediately from the network system to any user station. This facility is not exclusive to networked units but is enhanced by the speed and capacity of the larger shared system. The amount of system based software available on recent computer releases obviates the need for many software purchases and creates a quite different image of what a computer is.

3. Computer to Computer Communication

a) Electronic Messaging and Conferencing

It is not clear what purpose electronic messaging will play in education. Some people envisage students communicating between schools on a regular basis and teachers using the mail system to seek information on course content or teaching strategies. While the capability may exist, these interactions will not happen on a regular basis over time unless effort is put into conditioning teachers to this type of use and structuring student communications so that there is a clear purpose and some sense of closure on the activity. These more formal conferencing situations in which classes, small groups or individual students could participate are the object of much current study for business and university communities. The recent work by Keisler et al (1984) focuses on the behavioural and societal effects of this form of communication and identifies issues relevant to educational applications.

No doubt these facilities will be used by special students or in special situations. For example, handicapped students can now communicate directly and frequently either to others in the same restricted situation as themselves or to non-restricted students with whom they share interests and activities.

b) Networked software

A most exciting result of the spread of L.A.N. in classrooms is the prospect of networked software. Networked software involves the simultaneous use of a program by more than one user on a network in such a way that they can share the same screen or exchange messages among the group members. (Dwyer, 1981) While the strategy sounds merely intriguing at first, consideration of the applications of such a methodology opens new horizons for classroom use of computers.

The value of group work in education has been repeatedly demonstrated over the last fifty years and few would argue its benefits as preparation for our society and the work place. (Johnson, 1981;Davis, 1969;Orbach, 1979) Computer management of group work environments and learning activities presents the opportunity for unique advantages:

- anonymity can be attained on the network if such a situation is appropriate to the nature of the topic

- each student can be held to task by restricting certain capabilities of task steps to certain terminals
- messaging provides greater probability that points made will be 'heard' since they stay on the screen and arrive only one at a time
- previous peer group roles can be set aside to allow leadership to evolve from within the task providing students with a chance to demonstrate skills otherwise not utilized (Wilton, 1983).

Certainly, students should not be placed into networked group structures unless that environment extends or enhances the learning objectives. So little is known yet about how students will behave on networks or what the real characteristics of networked environments are, that predictions about the extent of future use would be premature. The development of experimental designs and related research to evaluate student learning behaviour in networked environments is currently underway.

Implications for Educators

Many of the current uses of computers in classrooms are enhanced versions of already accepted learning strategies. Networks create the opportunity for activities which are not used in existing school settings; but, which seem to have educational potential. Implementation of each of the applications presented here poses some special considerations.

Teachers

- The individual teacher may become less involved in the first level of software selection and will concentrate more on timing and preparation for the use of materials selected elsewhere and available on the network.
- New responsibility structures may emerge for teachers who manage network systems.
- In-service courses about computers will require new curricula with expanded emphasis on i) data bases - their structure and use, ii) management of class files on a disk system, iii) computer communications, and iv) the integration of large scale software packages such as courseware or unitware to curricula.

Students

- Full access to computer facilities at each student station could change the way students view computers and will add new sections to computer literacy courses.
- Students will be introduced to new dimensions of the role of a computer particularly concepts and skills related to computers and communications.

Software Design

- In designing user interfaces for networked computers, there may be a tendency to transfer a computer-centered conceptual base to the strategies for new operating systems. Educators must become involved so that the network can be made transparent and the concept of a learner's work space be the focus of the interface.
- Software development tools for graphics and sound have become very sophisticated. The same advances are needed to produce tools for networking so that software developers can easily link student stations or access external networks.
- There is a need to convince software producers that their products should be able to be used on a network - perhaps, with a different pricing structure for networked versions.

Many of the ideas presented here can be be assessed only in the future; but, others have immediate implications for teacher training, hardware purchases and software design.

REFERENCES

Davis, J.H. Topics in Social Psychology: Group Performance. Addison-Wesley, Reading, Mass. 1969.

Dwyer, T.A. "Multi-Micro Learning Environments: A Preliminary Report on the Solo/NET/works Project".Byte, January, 1981, p.204-216.

Johnson, D.W. "Student-Student Interaction: the Neglected Variable in Education", Educational Researcher, Vol 10(1), 1981, p.5-10.

Kiesler, S., Siegel, J. and McGuire, T.W. "Social Psychological Aspects of Computer-Mediated Communication". American Psychologist, Vol 39(10), 1984, P. 1123-1134.

Orbach, E. "Instrumental Simulation Games as Systems of Communication". Journal of Educational Technology Systems, Vol 8(1), 1979, p.67-93.

Wilton, J.A. "Structuring Learning Networks". ECOO Newsletter, Vol 4(3) p.25-31.

Human Factors Concerns In Educational Software Design, Selection and Evaluation

Janis Morariu, Ph.D.
University of Maryland
Hornbake Bldg. - Rm4121G
College Park, MD 20740

This paper focuses on the possible effects of the new interactive technologies on learning. From a human factors perspective, the question of what is really occurring when a child uses a word processing program in writing instruction, for example, is explored. The application of results from human-computer interaction studies of adult learners is also discussed.

An idealized instructional systems approach to materials design requires the cooperative effort of instructional designers, content specialists, curriculum specialists, teachers, and production specialists/programmers in structuring and conveying concepts and tasks for successful learning (Gagne & Briggs, 1979). In their book, Principles of Instructional Design, Gagne and Briggs (1979) characterize this systematic approach as focusing on the learning of relevant information, concepts, and/or skills by specified learners, through an organized plan of agreed-upon goals and objectives. Rather than support this perspective of learning, however, far too many publishers of computer-based instructional programs are developing materials for a nebulous home/school environment, outside of a clearly defined curriculum, and without the frame of reference of specific learners. Furthermore, it is not unusual for the latest technology to determine the path of instructional design (Gagne & Briggs, 1979).

Rather than allow the technology to dictate the design without a clear understanding of its influence upon learning, it is time for educational technologists and researchers to step to the forefront to analyze the effects of interaction between the instructional medium, the message, and the learner.

THE MEDIUM, THE MESSAGE, AND THE LEARNER

For two decades the power of media has been characterized by the philosophy that 'the medium is the message' (McLuhan, 1965). From a contextualist perspective of cognition, it appears that the total environment (or context) in which information is experienced does influence the processes of information encoding and recall (Jenkins, 1974; Watkins & Tulving, 1975). In developing learning support materials, however, it is critical that the instruction focus on the relevant content and skills, rather than on the context through which they are communicated. Even when the focus of instruction is a computerized environment and language like Seymour Papert's (1980) LOGO, the fundamental question that educators and researchers must ask about the technology is: How does the medium structure (or restructure) the learning environment and influence what is learned? In conjunction with this question, the following issue becomes critical: Can the use of a particular medium of instruction be justified in light of its possible influence on the learners and the learning task?

An example of this type of analysis can be found in an exploratory study conducted by MacArthur and Shneiderman (1984). These researchers examined the processes and difficulties experienced among two groups of fourth to sixth grade students performing below grade level in learning how to use two different word processing programs in writing instruction. Both programs were designed for 'ease of use'. Results indicated that difficulties stemmed from the following:

- Lack of typing experience (e.g., difficulty with spacing and the correct use of the shift key for capitalization);
- Inefficient cursor movement (e.g., using left and right arrows to move the cursor through several lines of text, rather than using the up and down arrows; difficulty with the concept of line 'wrap around');
- Confusion with insertion and deletion functions (e.g., inserting blank spaces in the text in order to 'open up space' to insert letters, words or lines);
- Problems in updating files (e.g., the concept that saving a new version of a text file under the old file name erases the old file);
- Difficulty in moving and deleting blocks of text (e.g., determining where to place the cursor to insert a block of text).

From the MacArthur and Shneiderman (1984) study one can infer that a computerized word processing approach and the manual paper-and-pencil approach to composition are conceptually quite different. With learners who are novices at both composition and word processing, this conceptual difference may be a critical factor in deciding which approach to use with what class of learners.

In his book, Interaction of Media, Cognition, and Learning, Gavriel Salomon (1979) argues that "the essential differences between media are the ways they structure and convey contents - and not their technological differences in communicating contents, although these do exist" (preface, xix). Salomon (1979) supports the theory that each media form possesses its own unique set of attributes which, "when used as carriers of the critical information to be learned, call on different sets of mental skills and, by so doing, cater to different learners" (p. 9). On the part of the learner, cognition involves an internal information decoding and encoding process whereby information is symbolically represented in memory (Kosslyn, 1975; Paivio, 1975). If, as Salomon (1979) argues, each media form consists of its own unique symbol system for representing information, research must examine the interaction between the learner's developing information processing system and the system of information representation conveyed through various media formats. Salomon (1979) further proposes that different media systems encourage the development of unique and varied information-processing skills among learners. Not only is the message structured through the medium, then, but so too are the developing thought processes of the new learner.

RESEARCH AND THE ADULT LEARNER

Most studies of human-computer interaction are conducted with adult learners, many for the purposes of business and industry. Researchers in the field are analyzing the design of the human-computer interface (Carroll, 1983; Good, Whiteside, Wixon & Jones, 1984), including screen display (Bork, 1983; Penna, 1984), and human factors (Egan & Gomez, 1982; Mills & Weldon, 1984), to name a few. Although there is great temptation to apply the results from these often well-funded and well-designed studies to the design and utlization of educational microcomputer programs with children and adolescents, one must proceed with great caution.

The process of learning to use the computer for various applications among adults most often requires the learning of a new technology to perform an already mastered skill (e.g., writing with a word processor or accounting using a spreadsheet program where the user already knows how to write and budget). The design of computer tools for skilled users, then, often takes advantage of the traditional context in developing a metaphor for the computerized context (Clanton, 1983).

This use of metaphor in the design of interactive systems for adults becomes important in assuring ease of learning (Clanton, 1983; Shneiderman, 1983). Since adults already know how to use a spreadsheet or a rolodex file, for example, using these familiar contexts for the design of computerized systems may reduce the amount of new learning required (Shneiderman, 1983). Designers of computer programs for novice learners who are new to both the intellectual concepts and the technological tasks, however, may find few previous contexts from which to draw helpful metaphors beyond the textbook and workbook (Goldes, 1983). The potential problems, then, of modeling the design of a new interactive lesson on a familiar non-interactive format may mirror those faced by producers when the new technology of television initially served as a radio with pictures. One may lose sight of the unique power of the new technology.

CONCLUSION

The first stage in the study of human-computer interaction among new learners (e.g., 3 - 12 years) should be the analysis of how thought processes are shaped by the interactive technologies, rather than how a knowledgable user already thinks about a familiar task. Then, if the place of a promising technology such as computers is to be assured within the educational system, the successful integration of the technology must occur through an instructional systems approach to software design and utilization.

REFERENCES

Bork, A. (1983). A preliminary taxonomy of ways of displaying text on screens. Information Design Journal, 3(3), 206-214.

Carroll, J.M. (1983). Presentation and form in user-interface architecture. BYTE, (December), 113-122.

Clanton, C. (1983). The future

of metaphor in man-computer systems. *BYTE*, (December), 263-279.

Egan, D.E., & Gomez, L.M. (1982). Characteristics of people who can learn to use computer text editors: Hints for future text editor design and training. *Proceedings of the 45th ASIS Annual Meeting*, *19*, 75-79.

Gagne, R.M., & Briggs, L.J. (1979). *Principles of Instructional Design* (2nd ed.). New York: Holt, Rinehart & Winston

Goldes, H.J. (1983). Designing the human-computer interface. *Educational Technology*, *23*(10), 9-15.

Good. M.D., Whiteside, J.A., Wixon, D.R., & Jones, S.J. (1984). Building a user-derived interface. *Communications of the ACM*, *27*(10), 1032-1043.

Jenkins, J.J. (1974). Remember that old theory of memory? Well, forget it. *American Psychologist*, *29*, 785-795.

Kosslyn, S.M. (1975). Information representation in visual images. *Cognitive Psychology*, *7*, 341-370.

MacArthur, C., & Shneiderman, B. (1984). *Remedial reading students difficulties in learning to use a word processor: Implications for design*. Unpublished manuscript. University of Maryland, Institute for the Study of Exceptional Children and Youth, College Park.

McLuhan, M. (1965). *Understanding Media: The Extension of Man*. New York: McGraw-Hill.

Mills, C.B., & Weldon, L.J. (1984). *Reading From Computer Screens* (Report No. CAR-TR-94). College Park, MD: University of Maryland.

Paivio, A. (1975). Perceptual comparisons through the mind's eye. *Memory and Cognition*, *3*(6), 635-647.

Papert. S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. New York: Basic Books.

Penna, D.E. (1984). The use of colour graphics display and touch screen to help naive users understand and control a multi-function computer system. *Interact 84*, 66-70.

Salomon, G. (1979). *Interaction of Media, Cognition, and Learning*. San Francisco: Jossey-Bass.

Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *IEEE Computer*, *16*(8), 57-69.

Watkins, M.J., & Tulving, E. (1975). Episodic memory: When recognition fails. *Journal of Experimental Psychology: General*, *104*, 5-29.

An Analysis Of Instruction Improvement Projects Utilizing Computer Resources

Frank Gillespie
Office of Instructional Development
The University of Georgia
Athens, GA 30602

The session will present an analysis of instruction improvement projects conducted at The University of Georgia. It will provide a comparison of projects using and not using computer resources and will illustrate how computer technology can be used to support or improve instruction. Comparisons will be made of instructional problems addressed by the project directors, the types of support they requested, the innovations they employed, and the products they developed.

Each year The University of Georgia's Office of Instructional Development provides support for faculty members who wish to develop new approaches to teaching. During the past two years a total of 25 projects have been funded, and of the 25 funded projects, 16 involved the use of some type of computer resource. The improvement projects involved undergraduate and graduate students in both traditional academic courses and professional programs.

During the funding period, faculty members from seven professional schools and seven academic departments within the College of Arts and Sciences submitted proposals that were judged by a campus-wide committee to be worthy of funding. Selection criteria included the number of students involved, the project rationale, the level of department commitment and originality. The College of Education was awarded five projects, of which four related to computers. The School of Environmental Design was awarded three grants, with two relating to computing. The College of Veterinary Medicine had three projects funded and two involved the use of computers. The College of Agriculture had two grants, both using computers. The School of Forest Resources was awarded two grants and one involved the use of computers. The School of Journalism was awarded one grant that utilized computers. Two grants were awarded to the psychology department and both involved computers. The departments of botany and legal studies each were awarded grants that used computers. The department of history, physics, political science, recreation, and romance languages were awarded grants that did not involve computers.

The improvement projects addressed a wide range of instructional problems. Five of the project directors submitted projects designed to simulate real events and three of these projects employed computer resources. Six of the projects focused on the teaching of specific concepts and four of these projects utilized computers. Of four projects designed to teach specific procedures or skills needed in a particular discipline, three utilized computer resources. Two of the projects were concerned with the management of information within their disciplines and both employed computer technology. Two projects were concerned with the teaching of problem solving skills and one employed computer capabilities. Three projects were concerned with the use of new tools in their discipline and all used computers in some capacity. Three projects were concerned with the development of faculty awareness and one employed computer capabilities.

While all the projects were designed to improve some aspect of instruction, the project directors employed a variety of computer capabilities and traditional or innovative instructional techniques to implement their projects. Of the sixteen projects utilizing computers, four were interested in the capabilities of interactive videotapes or videodiscs. Three project directors were interested in acquiring or developing computer assisted instructional materials to achieve their project objectives. Two project directors were interested in developing computer-based psychological tests in order to implement their projects. Two project directors were interested in utilizing data management systems and one employed a spread sheet application to achieve project objectives. Two project directors selected computer assisted design or graphics packages to meet project objectives and one utilized a plotter to produce anatomy illustrations. One project director developed a computer-based taxonomic key to achieve the objectives of his project.

Of the nine projects not involving computers, two utilized print material for the development of learning modules, three produced and utilized videotapes, one utilized specialized head phones, and one involved the production and use of slides. Two projects involved faculty participation in workshops or conferences to achieve their objectives.

The total budget for the 25 projects was $52,958. The total for the 16 projects involving computers was $32,393, with an average cost of $2,025 and ranging from a low of $788 to a high of $3,035. The total for projects not involving computers was $20,565, with an average cost of $2,285 and ranging from a low of $1335 to a high of $3,630.

The major expense for computer related projects was for equipment which involved 45.12% of the total computer project budget. The variety of equipment employed in the projects was extensive. Although the majority of the projects utilized microcomputer capabilities, two of the projects were conducted with the use of time share systems. Other projects employed procedures for interfacing with various types of video devices and laboratory equipment. Video production consumed 18.66% of the budget. Programming support was 17.51%. Costs for software were 11.62%. Release time costs were 6.17% and costs for supplies were 0.91%.

The major expense for non-computer projects was faculty release time which involved 35.89% of the non-computer project budget. Video production costs were 22.85%. Cost for equipment for these projects was 18.31%. Travel costs were 14.10%. Supplies were 4.77% and consulting fees were 4.08%.

A wide variety of instructional materials and products were developed as a result of the instructional improvement grants program. Two of the directors of computer related projects produced interactive videotape simulations; one relating to the administration and assessment of an informal reading inventory and one relating to classroom behavior management. One project director produced an interactive videodisc to provide faculty members with an introduction to academic computer applications. The director of another project acquired a biology videodisc and produced an interactive computer program to use with it. Two projects involved the development of data bases, one containing information on 150 veterinary pathology cases and another relating to characteristics of 150 species of trees. Two sets of psychological tests were developed and provided undergraduate and graduate students with experience with new testing techniques. CAI lessons in the areas of statistics and risk management were developed and a series on news writing lessons were installed on an instructional computing system. Two project directors developed specialized computer applications to support their instructional programs. One produced a spread sheet utility to teach animal nutrition and another developed a taxonomic key for classifying fungi. Another project director acquired a plotter and developed procedures for producing anatomy illustrations relating to veterinary medicine. Two directors acquired computer graphics software and employed them in courses in landscape architecture.

The nine projects not utilizing computer resources also acquired or produced a variety of instructional materials. Three sets of instructional videotapes, to support courses in physics, science education and recreation were produced from these projects. One set of slides relating to landscape architecture was also developed. A new course in wildlife habitats was developed and learning modules on American legal history and global security were produced. A collection of head phones was acquired to support courses in veterinary cardiology. A faculty workshop on new techniques in teaching languages was also conducted.

The instructional improvement grants program appears to be effective in stimulating the use of technology in a variety of academic situations. The value of the grants program in improving undergraduate and graduate instruction has not been fully determined. Although project directors were required to provide detailed descriptions of project implementation strategies, and problem resolution techniques, a need for more formal reporting procedures has become apparent. Methods used to conduct internal and external evaluations of project effectiveness continue to be refined, although some improvements in learning have been identified. Procedures for funding decisions, project management and project assessment will continue to be modified during future funding periods.

Diary Of A Computer Resource Teacher - Year One

Jane E. Matthews
North York Board of Education
5050 Yonge Street
North York, Ontario M2N 5N8

My classroom is equipped with six Icons, six PETs, nine Apples, a variety of peripherals, and a wealth of software. As computer resource teacher I have the responsibility of integrating computer-related activities with the ongoing classroom programs of all junior division students in the school. The key objective of the project is **evaluation** -- evaluation of hardware, of software, of methodologies, and of student learning.

The decision of the North York Board of Education to establish two regional computer classrooms was made early last spring. When I was hired for the role of computer resource teacher in the Western Region Classroom, I was presented with a laudable rationale and a formidable list of objectives.

Rationale for the facility:
to generate and test applications of computers with children; and to help determine the future direction of computers across the curriculum.
Objectives:
to answer these questions about the impact of computers in schools:

1. What is the effect of word processing on creative writing skills?
2. How may certain problem-solving skills be enhanced through intensive computer use?
3. What is the feasibility of creating a co-learning environment involving teachers and students with Logo?
4. What behavioral changes in the social, emotional, and cognitive domains can we observe in students given generous daily access to computers?
5. How might the computer give students control over their learning and influence their self-esteem?

Designing the process for fulfilling those objectives is now in its early stages. There is no precursor to this innovative situation and there are no existing guidelines for setting up a computer classroom for the purposes outlined above. This paper will detail the steps taken at this writing (January 1985) and will outline the plans for the rest of this academic year.

In September, 1984, a successful start for the computer classroom required: support from assistant superintendents, involvement of the school principal and caretakers, positive interest from classroom teachers, a suitably equipped room, hardware, software, and the opportunity for the students to use the computers.

In spite of the support of senior administration and the collective efficiency of the physical plant and purchasing departments, the facility as we use it today was far from ready at the end of the summer. The computer classroom was two rooms -- a dreary guidance office and an emptied stockroom still displaying damage from a break-in last June. Institutional green walls, crumbly bulletin boards, an old slate blackboard, four electrical outlets, and burned-out fluorescent tubes greeted me on September 4.

I was also greeted by fifty-five eager students who knew that computers were to play a special part in their lives at school this year and so the temporary computer classroom was set up -- in the second gymnasium. Large tables and power bars were placed against the four walls. One Apple and three PETs belonging to the school and two borrowed Apples were moved into the cavernous space. A shag carpet and chesterfield pretended to make the place look cozy.

Being new to the school, I needed to get to know the children. We talked about computers in general and about the Computer Classroom Project in particular. I emphasized their role in helping educators to discover how computers could help people to learn. They were told that they would pioneer some of the special uses of computers because our Board of Education had decided to buy certain equipment and programs to be tested in our school. The children have known since the beginning that they would participate in the evaluation and sharing process and are proud to be part of that.

Although we had grand plans and a sizeable budget, we still had to begin at the beginning. A number of public domain programs were on hand and a few pieces of commercial software had been delivered. We had three PETs and three Apples and the children came in groups ranging in size from seven to fourteen. We began using Storywriter on the PETs and Gertrude's Puzzles on the Apples. It was interesting to observe the continuity of activity from day to day as the children logged what they had accomplished and planned their future work.

While we were still in the gymnasium, a shipment of new Apple IIe's arrived, we acquired four more PETs, and the first two Icons were delivered. Printers appeared, often weeks ahead of their interfaces. Miscellaneous software showed up, followed by powerpads and mice, a modem and a turtle. Meanwhile the construction continued in the classroom. At the beginning of November we moved in.

Our Computer Classroom is a standard classroom divided into two rooms with a window in the wall between. Six very large tables, four study carrels and a built-in counter hold twenty computers, four printers, and the file server in a functional arrangement. Two other tables in the centre of the larger of the two rooms provide a work surface and a place to meet. A large monitor for demonstrations is mounted in a high bulletin board. We have cupboards to store our software and documentation, file cabinets for student work, and custom-built display space for books and magazines. The slate board was replaced with a (dustless) white board and extra bulletin boards were added. The old guidance office telephone is used with a modem. The phone has been modified to prevent interruption from other phones in the school when in use without the expense of installing a dedicated line. After several months of planning, designing, requisitioning, and rearranging, the Western Region Computer Classroom is in place.

If it were an ordinary computer lab, perhaps it would house many copies of one computer but as a test-site for the growing variety of hardware and software, this computer classroom contains an eclectic mix of equipment. That means that each group of students will come in and proceed to work at different activities. Some will do word processing on a PET or an Apple or an Icon. Others will create graphics using a number of different input devices. Some detective teams will continue with their "Snooper Trooping". Others might create a "Word Search" puzzle using the vocabulary they are studying. Like many elementary classrooms, this room contains many activity centres, but here it is computers which provide the environment for student thought and creativity.

The two major tasks we completed during the fall term were to set up the physical facility and to establish among the students independent and responsible attitudes towards the use of the software and hardware.

One of our next tasks will be to involve our Board Research Department in designing and undertaking research projects which will help us to answer some of the initial questions about the educational impact of microcomputers.

The other major task is to explore particular pieces of software with students and share what we learn (through workshops or publication) with teachers throughout the Board. Collecting that information will involve a simple sequence of steps which evolved from the work we did in the fall.

For any given piece of software:

1. The software is chosen and purchased.
2. The resource teacher becomes familiar with the software.
3. The students are introduced to the software.
4. The resource teacher observes student use of the software.
5. The resource teacher shares information about the software with the classroom teacher to cooperatively develop curriculum-related applications of the software.
6. The students, now competent users of the software, are challenged to apply it to a task based on their classroom work.
7. The resource teacher and the classroom teacher monitor and evaluate the students' individual and collective use of the software for the challenge(s) given.
8. The students evaluate their use of the software.
9. The students may have the opportunity to teach others how to use the software or to apply it to self-directed projects.
10. The results will be summarized to be shared with other teachers.

We plan to complete this process for certain pieces of software for the following applications:

Logo,
word processing,
information management,
spreadsheets,
electronic communications,
graphics/animation,
music generation,
logic puzzles,
simulations.

Later, the project will expand to involve students and teachers from selected schools throughout the Western Region of North York. The role of the resource teacher will include working with teachers in other schools to implement computer projects. The cumulative evaluation of such projects in other schools will provide valuable data for the decision-makers choosing the future directions of classroom computer use in the North York Board of Education.

On-Line Data Bases: A School Project

Michael S. Levinson and J. Andrew Walcott
Rutgers Preparatory School
1345 Eastern Ave.
Somerset, NJ 08873

Telecommunications and the use of On-line Databases are playing an increasingly important role in the acquisition and use of information in all aspects of society. Education must begin to prepare students to utilize this very important tool in the pursuit of knowledge. Rutgers Preparatory School, a small independent school (Kindergarten - 12), developed a pilot project in which students and faculty used an On-line Database Service (DIALOG) in the development of Science research projects.

ON-LINE DATABASES--A SCHOOL PROJECT

Information Services are tied to the future of education and educational research. As the amount of information and organized data in the world continues to grow geometrically, we believe that it is important that educators and students learn to use this new tool in the search for, and acquisition of information. These skills will play a significant role in the life of academic professionals.

Mr. Andrew Walcott - Director of Media Services 7 - 12 and Mr. Michael Levinson - Director of Computer Services at Rutgers Preparatory School, a small independent school in central New Jersey, received a $1000.00 grant from the school to develop and initiate a pilot project utilizing On-Line Databases in the high school classroom. Since we had Apple IIe microcomputers, we purchased a Hayes MicroModem IIe and SmartCom Communications software. The micromodem installed directly into the Apple IIe and the SmartCom software was extremely easy to use. In addition, a dedicated telephone line for telecommunications was installed in the library/media center and in the computer science lab. The cost of the modem, software, and the telephone line was about $350.00. In surveying the various information services available as listed in "OMNI-Online Database Directory"[1] it was decided to use DIALOG. DIALOG consists of over 200 databases that span the range from the "Academic American Encyclopedia" to the "Zoological" Record". It seemed to provide the most comprehensive spectrum of subjects available for educational research. There is no subscription fee, no minimum service charge, and you pay only for the time and services used. Also, DIALOG offers complete training services in the use of the DIALOG system and query language.

During the summer of 1984, Mr. Walcott and Mr. Levinson took the DIALOG training course and developed an outline of the pilot project. That fall a meeting was held with the Science, English, and History Departments during which an overview of information utilities, computer networks and a session with DIALOG was presented. The goal was to select a teacher in each of those departments who would be willing to utilize DIALOG with their students on a research project. Three faculty members volunteered; two science teachers and one of the history staff members. The first project to be initiated was by Ms. Marylin Bell, Physics teacher. Her students, in two different classes, were to undertake a research project resulting in a term paper on Alternative Energy Sources.

Introducing technology into the school is a difficult task. There are many who tend to see it as a short-lived panacea that will go the way of the new math and teaching machines. If an addition to the educational program is to be made and is to be practical, it has to be integrated into the schedule and the understandings of those who will be affected by it.

In our pilot program, we were interested in the impact of on-line database services on: teacher, student, and media specialists' time; the quality of information relative to cost and time involved; and teacher and student attitudes to the use of on-line research.

One of the classes involved in the project was to use on-line searching while the other made use of traditional library search techniques.

The on-line group received an overview of databases and search strategies and procedures. The teacher worked with the class to help guide them in writing search request forms, which were to be presented to the media specialist. The initial requests fell into three groups: requests ready for search; those that needed rewriting in terms of database language, and those that needed background research in the library. The latter two groups met with their teacher two or three times during class time to complete statements for searching. The students using traditional methods spent their class time in the library doing independent research and worked with the teacher as necessary.

Search statements were entered into DialIndex which searches across several similar subject databases for selecting the database most useful to search in. The media specialist entered the students' search statements into the system and returned the output to the students. The media specialist spent half to three quarters of an hour per day conferring with students or doing searches.

The students used their bibliographies to find resources and write their papers. While the time spent for the teacher and students was approximately the same in both groups, the media specialist spent significantly more time with the group using electronic methods.

We restricted student access to the computer because DIALOG is costly and can only be used by trained people. Also access to the service is via a password that needs to be kept secure. The students wanted to do the on-line searching themselves. They cited two reasons: they felt, 1. detached from the process and 2. a need to make on-the-spot decisions rather than be delayed through an intermediary. We concluded, therefore, that on-line searching should be done by a trained teacher who guides the students as they proceed through the process.

We found that most of the information, although of high quality, was inappropriate for this particular group of searchers. The bibliographies gave sources that were too technical for most students. Our first try did not yield general scientific database that was on a high school level. It was hard to tell if the searches were performed to optimum level or, if information was missed or overlooked. Only continued practice and experience can provide the answers.

We estimate the cost of our searches to total between $50 - $75 for this project. Also, we did not access full text articles -- only abstracts. The cost of retrieving a full text copy averages $7 - $10 each. These expenses could not be justified beyond the experimental stage, so our next two projects in history and foreign language will help complete our evaluation.

Although our first project did not satisfy our expectations, we believe that when four conditions are met, on-line searching will be productive in schools: 1. student access to the computer and database, 2. cost control, 3. more generalized educational databases, and 4. access to full text. We see a need for educational consortums on state or national levels to fund and produce curriculum oriented databases.

We did observe a major difference in the student-teacher relationship between the two search processes. Traditionally students narrow down their topic from a body of information that is available locally. They learn as they go and let the search take shape based on what they find. In contrast, on-line searching demands that they form a specific, well-defined concept and then retrieve that specific information. The on-line process stimulates interaction with the teacher as a subject specialist and is conducive to careful preparation and thought. The intellectual process of classifying subject matter and expressing the search in the precise language of the discipline is a strong educational justification for the process. Given the conditions above, we believe that on-line searching can add an extra dimension of intellectual power to the scholastic lives of our students when combined with traditional sources.

....

1. OMNI-On-Line Database Directory Edelhart and Davis, Collier Books. MacMillan Publ. Co., NY., 1983.

Computers And Art

Larry Noonan
Metropolitan Separate School Board
787 Strouds Lane
Pickering, Ontario L1V 3M2

Using computers in art education has met with various degrees of acceptance. Some say that computers have no place with such a personal creative form of expression. Others claim that there is always room for new methods and ideas.

The fact is that, like it or not, computers are being used in a variety of ways to 'enhance' the teaching of art in elementary and secondary schools, and in universities and community colleges. Some of the ways that the computer is used are:

- in the teaching of art appreciation
- as a tool for the teaching of the various elements of artistic design
- as a new artistic tool or media with which to create

Traditional Art

The first two ways mentioned above call upon the computer to help teach about the so called traditional art forms. One method that is often used, since the graphics on a computer are limited, is to interface the computer with a videodisc player or slide projector. For the study of art, the slide projector is most often used.

This involves connecting the slide projector to the computer. The slides are of actual paintings or sculptures. The computer gives instructions to the student or asks questions. The movement of the slide projector and the video screen display are controlled by the responses of the student. The slide projector will move forward when the child answers questions correctly. Likewise, the slide projector can move in reverse to review certain facts when the child is experiencing difficulty answering the questions. This is a crude form of computer assisted instruction (CAI) because it progresses in a linear direction either forward or backward. (It is possible with a more expensive slide projector than we normally find in our schools to have the computer move the projector to any slide that is needed. This would be closer to a true CAI technique.)

Figure 1 shows the circuit diagram for an interface that can be used to connect a Commodore 64 to a Kodak slide projector.

Figure 2 is the driver program for the slide projector. This program contains two subroutines that can be accessed from the main program.

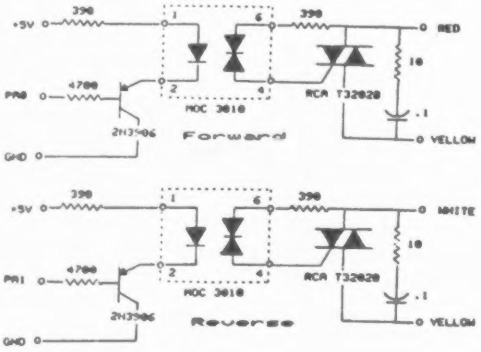


Figure 1 - slide projector
interface circuit

BEST COPY AVAILABLE

```
5 REM M.S.S.B. TORONTO. UPDATED SEPT.84
8 REM FOR USE WITH 4032/8032/C64/VIC20
10 PRINT"J"
15 M=59459:N=59471:REM 4032/8032
20 FEM M=5E579:N=56577:REM C-64
25 REM M=37138:N=37136:REM VIC 20
30 POKE M,3: POKE N,3
40 PRINT"[redacted]   SLIDE PROJECTOR CONTROL"
45 PRINT"[redacted]        DEMO PROGRAM"
50 PRINT"[redacted]F      ADVANCES MAGAZINE"
60 PRINT"[redacted]R     REVERSES MAGAZINE"
70 GET A$: IF A$="" THEN 70
80 IF A$="F" THEN GOSUB 200
90 IF A$="R" THEN GOSUB 300
100 GOTO 70
200 REM ADVANCE MAGAZINE
210 X=PEEK(N)
220 X=X AND NOT 1
230 POKE N,X
240 K=TI
250 IF TI<K+20 THEN 250
260 X=X OR 1
270 POKE N,X
280 RETURN
300 REM REVERSE MAGAZINE
310 X=PEEK(N)
320 X=X AND NOT 2
330 POKE N,X
340 K=TI
350 IF TI<K+20 THEN 350
360 X=X OR 2
370 POKE N,X
380 RETURN
READY.
```

Figure 2 -driver program

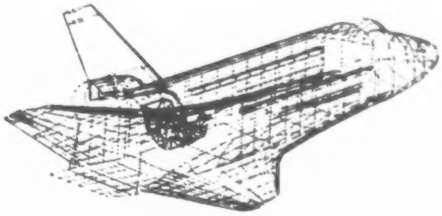


Figure 3 - Line drawing of the shuttle

By using the slide projector in conjunction with the computer, the student can see reproductions of real works of art and be directed to specific observations.

The New Media

Another way to use the computer in art is by using the technology as a new media to create works of art that are different from the traditional forms. The techniques for this range from very primitive to the very advanced.

One advanced system is called Genigraphics. This system, from General Electric, allows artists to draw pictures, store them on disks and photograph them in full colour. The results of this technique are often very realistic and can be mistaken for photographs of real subjects.

Most schools, with the exception of universities and community colleges, will not be using such advanced methods in their computer art classes. At the elementary and secondary school levels, microcomputers and artistic software packages are commonly used to create art. Line drawings, such as the one shown in Figure 3 can be created on the Apple and C-64 and are often used in computer aided design classes.

Popular programs like KPaint, IPaint, Doodle, Print Shop, PC Paint, Logo and Flexidraw can be used with a joystick, light pen or graphics tablet to create very acceptable pieces of art. These and other programs are used in Junior Kindergarten right up to secondary schools. They can be used to:

- draw pictures
- design posters, cards and banners
- design a piece of art and study colour combinations on the computer before using more traditional media, such as pastels or paint, to make the picture
- study shapes and how they can be combined to create pleasant designs
- teach the basics of point, line, shape and colour combinations

The final results of the student's creative work can be printed out on a graphics printer. Figures 4, 5, 6 and 7 show examples of printouts. These can be displayed as is, or

BEST COPY AVAILABLE

water colours, paint, crayons or other media can add colour to the final piece.

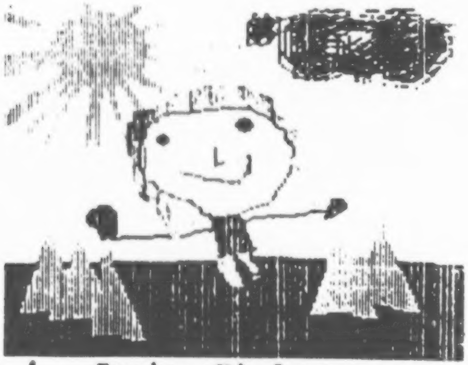
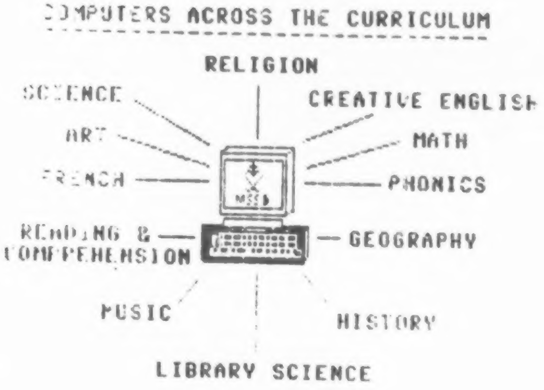
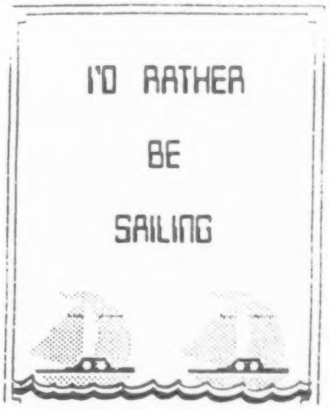


Figure 4 - Junior Kindergarten child's picture done with a Koala Pad



COMPUTERS ACROSS THE CURRICULUM
RELIGION
SCIENCE
CREATIVE ENGLISH
ART
MATH
FRENCH
PHONICS
MSSB
READING & COMPREHENSION
GEOGRAPHY
MUSIC
HISTORY
LIBRARY SCIENCE

Figure 5 - Line drawing done with Doodle



I'D RATHER
BE
SAILING

Figure 6 - Front of card created on Print Shop

Some graphics programs allow the final product to be printed out on a colour printer. If a colour printer is not available, a photograph or slide can be taken from the colour monitor. Enlargements can then be made and displayed.

Summary

The idea that the computer has no place in art will remain with many people. However, the computer can be used to help many of us appreciate traditional art while at the same time give those of us who want a new media with which to work and create art.

References:

- Computer-Slide Projector Interface - Ted Evers, MSSB, Toronto, Ontario
- Computers in Art Education - Resource Document, MSSB, Toronto, Ontario
- The Elements of Design - Resource Document, MSSB, Toronto, Ontario
- KPaint, Koala Pad, IPaint, Doodle, Print Shop, PC Paint, Logo and Flexidraw are all registered trade marks.

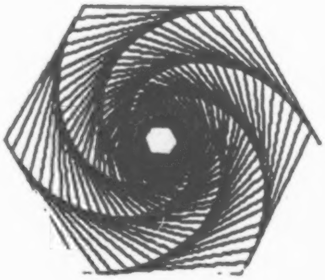


Figure 7 - Logo design

Computer Applications Across The Business Education Curriculum

F-G. Anrep-Matthiesen
Applewood Heights S.S.
945 Bloor Street, E.
Mississauga, Ontario

There are no bounds or borders as to how computers can assist and enrich the Business Education Curriculum.

When computers are refered to, Data Processing or drill and practice normally come to mind, but these are only the beginning. The most exciting prospects of computer use exists in all the other areas of the Business Education curriculum. Wherever you look the prospects are limitless.

The main areas that will be examined in this discussion are: Marketing, the school store assistant, the modern office (practice), Applied Accounting and other accounting applications, business simulations, Law case studies, and research using data bases.

The most interesting applications in marketing involve promotions and advertising. Using graphics packages students can design advertisements and prepare promotional layouts that involve both diagrams and different lettering styles. By varying the design different options can be examined without having to draw each by hand.

The accounting functions can also be demonstrated. Calculating prices, profit, mark-up, discounts, and all the other processes related to the operation of a business. There are several good simulations available that allow the students to actually operate a business of their own.

Included in the marketing program of most schools is a school store. This can be a very time consuming effort. If the software available is used to its fullest the teacher can be freed from the menial record keeping tasks to perform the managerial duties required. A combination package that includes at least a spread-sheet and data base system would be best for this.

The modern office is another subject area where great changes are taking place. The modern secretary needs to be able to operate more than a typewriter and copy machine. Many offices, both large and small, are using electronic word processors, data base file storage, electronic mailing, and computerized record keeping.

In Applied Accounting the students work on actual cases as they would be done in industry. Spread sheets used in conjunction with graphing and trend creating programms allow the students to do these tasks. The spread sheet can also be used when creating and working with worksheets. This would allow students to examine how changing the values in the starting trial balance or the adjustments affect the net income.

In some courses the students would benefit greately if they could work with realistic problems. There are now a number of business situations that can be simulated using the computer. These can be as simple as operating a lemonade stand or as complicated as running a major industry in competition with other students.

BEST COPY AVAILABLE

The computer has the additional advantage that if the simulations available do not suit the teacher then they can be adapted or, if the needed skills are available, custom simulations can be designed for the situation at hand.

In some courses it would be useful to have cases on file that students can access for research purposes. This can easily be established using any of a number of data bases now available. The students would then be learning the subject in question and how to find material relating to it using current technology.

As has been mentioned earlier there are two ways to use the computer in performing these tasks: purchase a specific program for the application wanted or have someone in the school design and write a new program or customize an existing applications program. Students from the senior computer classes will often be more than happy to tackle a project like this.

For many of the applications refered to here commercial utility programs adapted for the given situation are the simplest solution. Spread sheets such as Visicalc, Lotus, and Calc Results are exceptionally versatile for record keeping and accounting applications. Packages such as Lotus and Appleworks expand this flexibility. Applications requiring file access can use packages such as PFS and DBase II.

For applications such as creating advertising for a marketing class or doing other graphic work there are not only utilities but also several languages designed to let the user draw and print in varying fonts. The utilities can be as basic as EDU Paint or as complex as Print Shop. When one examines the graphic languages that are available the variety ranges from LOGO to GRAPHForth. In addition to these choices there is also hardware in the form of tablets, light pens, joy sticks, roller balls, and mice that can be added to make the drawing easier.

There are many programs available for office practice training. The student can learn how to type using programs similar to Typing Tutor. Word processing (or text editing) can be learned on Word Star, Appleworks, Volkswriter, or numerous other packages that are available. Many of these can also be incorporated with a data base and spread sheet presenting the student with a complete office system.

These are some of the applications that will challenge students in the business program and prepare them for work in the electronic environment that is becoming the norm in business.

Strategies For Developing Faculty Computer Literacy

Sister Ignatia Frye and June A. Parsons
Northern Michigan University
Marquette, Mi 49855

ABSTRACT

Learning to use personal productivity applications software is a significant step in developing faculty computer literacy. These applications can be useful in the faculty member's professional career: in research, writing, and class management. They can also serve as building blocks for defining computer applications unique to the faculty member's field of expertise.

INTRODUCTION

Beginning in the spring of 1984, Northern Michigan University offered its faculty, administrators, and staff intensive in-service training on computers, emphasizing personal productivity applications software. Training modules included in this program were word processing, spreadsheet design, and database management. A BASIC programming module was also offered as a vehicle for increasing understanding of computer terminology and architecture. The faculty in-service training utilized the same software as Principles of Data Processing and Information Systems which is the basic computer literacy course offered to students.

Faculty members selected the module or modules in which they wanted to participate. Each module was scheduled for 12 hours of instruction, meeting four consective days for three hours each day. Due to high demand for these programs, the session offered in the spring of 1984 was offered again in the late summer.

EQUIPMENT AND PERSONNEL

The faculty in-service training was sponsored by the university Office of Human Resources. Instructors for the course were selected from the Department of Management, Marketing, and Computer Information Systems and the Department of Mathematics and Computer Science. The instructors all had extensive background teaching beginning computer students.

The university Educational Computing Facility containing 25 microcomputers was the primary instructional site. Copies of SuperWriter, SuperCalc2, and dBASE II were provided for each microcomputer through special licensing agreements with the publishers.

The in-service training was open to all faculty, staff, and administrators of the university. Released time was granted to all the participants. An analysis of data showed a great diversity in department affiliation and job description of the participants.

METHODOLOGY

The availability of one microcomputer for each participant provided the opportunity for a completely practical or "hands-on" laboratory approach. Methodology was also shaped by a perceived need to create a highly structured presentation emphasizing the underlying principles and concepts of the applications.

The methodology was designed specifically for a well-educated adult population. The approach, using a step-by-step logical progression, can be outlined as follows:

I. A problem designed to be solved by a specific applications program, such as a spreadsheet, was presented.
II. The step-by-step procedure for solving the problem was presented. For each step in the procedure:
    A. The general concept was explained
    B. The specific procedure was described
    C. The procedure was implemented on the computers
    D. Questions were answered
    E. Another similar example was provided for the participants to do individually
    F. Final questions were answered
III. After all the steps had been covered, the origi-

nal example was examined as a whole. At this time a review of the procedural sequencing was appropriate.

IV. A sample problem similar to the original example in (I) above was provided for individual experimentation. Often, participants were encouraged to create their own tasks based on suitable applications in their field of expertise.

Instructor preparation for the in-service training required the selection of appropriate examples and the creation of related documentation materials including transparencies, problem descriptions, and command references.

EVALUATION

Evaluation tools provided interesting perspectives on the in-service training. The sessions were rated highly in terms of perceived current and future needs. The structure and content of the training were also given high marks. Follow-up questionaires are currently being evaluated.

Instructor evaluation of the course indicated difficulties in dealing with a wide variety of previous computer experience. Though the in-service training was specifically designed for computer neophytes, many of the participants were quite skilled in computer operation. Evaluations from the participants also indicated problems in this area. In the late summer session, an attempt was made to offer "pre-training" which covered basic computer concepts for those participants who had no prior computer experience. The effect of this strategy was somewhat diminished by the fact that several beginners were unable to attend this session.

DISCUSSION

The faculty in-service computer training at Northern Michigan University must be considered a success. It has exposed many faculty, staff, and administrators to basic computer applications which can become useful tools in their work at the university. Theoretically, this knowledge can form a basis from which the faculty member can explore applications specific to his or her area of expertise.

A unique feature of the in-service training was the emphasis on software use. Analysis of hardware needs and the relative merits of different brands of computers was minimized. The training might best be described as a "Users Seminar" rather than a "Buyers Seminar."

In the future we would recommend a continuation of this introductory applications series for computer neophytes. For more experienced computer users we would recommend a series of short courses offered on specific uses of these applicaitons, i.e., Using the Word Processor to Produce and Revise Exams, Use of the Data Base to Manage Grading, and Use of the Spreadsheet in the Analysis of Case Studies. A possible new direction would be to offer a seminar which would assist participants in an analysis of possibilities for effective computer use in their specific field of expertise.

CONCLUSION

The faculty in-service training has been a significant step in developing computer literacy among the faculty and staff of Northern Michigan University and may be used as an effective model for implementing similar training programs at other universities.

Minnesota Information Technology Learner Outcomes

Dr. Harold G. MacDermot
Minnesota Department of Education
550 Cedar Street
St. Paul, MN 55101

Minnesota has adopted the philosophy that a higher technical world creates new needs for students and requires a changed education system; one where students learn about technology while they use it to learn the content of other subjects. An "Information Technology Learner Outcomes" document has been prepared to provide school districts with direction to meet these new student needs.

The Minnesota Board of Education, on November 18, 1983, adopted a resolution which stated that every school district must offer an Information Technology Program. This paper describes the "Information Technology Learner Outcomes" which is the State Board's interpretation of how the minimum program requirements for Information Technology might be implemented. Statewide committees of educators provided valuable insight into what students should know about technology and how to use various technologies for learning. The general goals and learner outcomes they developed are appropriate for all students from kindergarten through grade 12.

The goals and outcomes are neither grade-level nor subject-area specific. They are intended to be interdisciplinary because it is our contention that students may learn with and about technology in all subject areas. The technology goals and outcomes may be covered within existing curriculum without creating new courses.

Thus, the underlying assumptions for this paper are that material about technology is best provided by including it along with the existing curriculum, and that the information technology curriculum needs to be flexible enough to adapt to rapidly changing technologies.

This paper describes the first of five sections which the complete "Information Technology Learner Outcomes" document will contain. This section identifies the necessary knowledge base for all students to have before they begin a more indepth study of computers, telecommunications, media production and robotics. It is recommended that this Technolgy Overview Section be incorporated into the existing curriculum by all school districts.

Nine broad strands make up the Technology Overview.

These strands are:

I. Technology Applications
II. Operational Procedures and Characteristics
III. Technology Assisted Problem Solving
IV. Group Processes in a Technology Setting
V. Social, Economical and Political Implications of Technology
VI. Aesthetic Aspects of Technology
VII. Ethical Considerations of Technology

VIII. Technological Futures
IX. Communication Implications of Technology

Each strand is divided into three levels. These levels correspond to Benjamin Bloom's "Taxonomies of Educational Objectives". Level A covers knowledge and comprehension, Level B contains application and Level C concerns higher-thinking, including analysis, synthesis and evaluation. The goals and learner outcomes are recommended for implementation by everyone. The sample activities are only suggestions as to how the outcomes might be accomplished. Strand IV, Group Processes in a Technological Setting is included on the following page to illustrate how the document is organized.

A higher technical world requires a changed education. In a challenging global information society, we need a different kind of education than we have had before. One of the changes in education will be a greater emphasis on creativity, higher-order skills, problem solving conceptual mastery, intellectual independence, the ability to express oneself clearly, and to work collaboratively with others, including people with different political, social and economic values.

Information Technology will certainly play a major role in the redirection of future education. Thus, a major caution must be that what is outlined here represents our best thinking today and that this outline must be subjected to systematic review and modification as technologies and learning materials improve.

Another assumption to keep in mind is that educational technology needs to be considered as only one of several educational tools and delivery systems available for learning and teaching--superior in some situations and inferior in others.

An additional assumption that needs to be highlighted is the belief that considerable effort must be made to ensure that students learn and use higher-level thinking applications of technology. We believe the potential of technology cannot be fulfilled unless students are taught problem-solving and decision-making skills that have real-life implications and uses.

A common mistake being made in many schools today is that the focus of instruction is on teaching computer literacy and/or programming. While the number of microcomputers available to students is increasing rapidly, computers represent only one segment of the technology field. Telecommunications media production and robotics should also be included as a part of the information technology curriculum.

Changes once occurred very slowly. Now, not only are the changes occurring at a rapid rate, the rate of change is accelerating to the point that information is doubling every eight years in most areas and, in some areas, such as microbiology and computer engineering, every 18 months. Technology in society is not a fad. Experts estimated that over $59 billion were spent on telecommunication resources devoted to the purchase and operation of computer-related hardware and expenses were to be $150 billion in 1983, an amount that is expected to increase to as much as $700 billion in 1983. Total expenditures in high technology areas were estimated at $400 billion in 1983. There is disagreement about how high future totals might be, since new applications of technology are being developed almost daily. Given those figures, however, it is obvious that business is recognizing the value of high technology and that developers and producers are responding to the demand for rapid changes in the amount and uses of technology.

IV. GROUP PROCESSES IN A TECHNOLOGY SETTING

GOALS	OUTCOMES	SAMPLE ACTIVITIES
LEVEL A - Knowledge and Comprehension		
1. Students will understand ways which working together with others using technology helps increase productivity.	a. Students will understand that working with another person to learn a new procedure will increase the speed at which both people learn.	a. In a computer lab, students will work individually for a defined period of time to solve a problem. They then work in pairs for the same time to solve a comparable problem. The class will compare the success of the two methods.
	b. Students will understand that working in a group situation is necessary to the operation of certain technologies.	b. Students will view a videotape which describes the interrelated jobs of each person involved in a video broadcast.
	c. Students will understand that helping another person to learn a new procedure will increase one's own knowledge and understanding.	c. Students will invite community members (parents, etc.) into the classroom and instruct them in the functions of the different components of a computer system.
LEVEL B - Application		
1. Student will be able to work cooperatively with others using technology.	a. Students will work together to maximize the effectiveness of using a technological tool.	a. Students will work in pairs at a computer learning station to write procedures using a simple programming language to solve a problem.
	b. Students will work together to use technologies that require planning and division of labor to produce a desired outcome. (By combining the individual strengths of the group, a synergistic effect will result.)	b. Students will work in a team to write, direct, stage, film, edit and present an orignal video production. Students will work in groups to create newspaper, radio or television ads promoting a software program they have used.
LEVEL C - Higher-Level Thinking (Analysis, Synthesis and Evaluation)		
1. Students will organize group efforts to use technology and will assess the ways technology increased the productivity of the work.	a. A student will select other students with the skills needed to plan, develop and document their data.	a. Using computers and video, a team of students will script, cast and produce a news documentary.
	b. Students will organize a project to use technologies that require planning and the division of labor to reach a desired outcome.	b. Students will direct the work of a team that writes, creates and edits an original video production.
	c. Students will evaluate the productivity of a team-based technology project and recommend ways to improve future efforts.	c. Having directed a video production, students will write an evaluation specifying improvements for future video projects.
	d. Students will use technology to assist in maintaining cooperative efforts, encourage group problem-solving and refine the service or product, based on team evaluation.	d. Students will plan and develop a business appropriate for a school setting (e.g., selling pencils, home-made crafts, items displaying the school name or mascot, etc.), using the computer for record keeping of meeting minutes, sales, inventory, work schedules, bank balances, student stock investments, etc. Students will graph profit or loss results to assess progress of the business.

BEST COPY

Rapid changes require institutions to take risks to implement significant changes. Minnesota's educational institutions must take risks and make changes if its students are to compete successfully in an economic and social structure with global implications. The emerging society requires the processing of vast amounts of information to solve problems and make quality decisions. There is an increasing need to know more, to know how to access vast amounts of information and to improve the way and the rate at which we learn. Technology offers educators a means to address these increasing demands on education. If educators accept the changes as a challenge rather than a crisis, and seize the opportunities provided, they can make education more exciting and productive than ever before. In Minnesota, the Legislature, the State Board of Education and the Department of Education are working together to encourage and support the public schools to implement significant changes in their use of educational technology.

"MacEnglish" - Using The Macintosh For Senior High English And Journalism

Tom King
St. Paul Public Schools
360 Colborne
St. Paul, MN 55102
and
Mark Parish
Madison Wisconsin Public Schools

Micro-computer technology typified by the year-old Apple Macintosh will have immediate effects on the way traditional English and Journalism classes are taught and the way in which schools plan, execute and publish student newspapers and yearbooks. Of course,more and more students will continue to use micros to learn to outline, write and rewrite. With the increased use of inexpensive high-quality laser printers, it can be expected that most elementary, middle and secondary schools will produce documents such as graduation announcements, student newspapers, yearbooks, and athletic programs entirely within the school.

The speed with which the microprocessor is changing lives stuns most people, but few are more stunned than those who are teaching or administering in public schools in the United States, institutions which have substantially unaltered their methods of operation, teacher-to-pupil ratio or fundamental course offerings in the last 100 years.

To these teachers and supervisors, the clouds of microprocessor-inspired-change hanging over the horizon seem overwhelming and threatening.

The microprocessor revolution seems initially to have resulted in American schools purchasing individually inexpensive microcomputers such as Commodore, Atari or Apple and placing them in lab settings where beginning students work on programming in Basic and advanced students experiment with Pascal, most often under the direction of math-teachers-turned-computer-teachers.

To many veteran scientists of the computer age, this rush to expose children to methodical classroom lessons in basic programming, and to create whole computer sections filled with fourteen year olds for whom "strings and arrays" are tangential interests, the promise of the computer is being seriously betrayed. Among those critical of this initial tendency is Alan Kay, the direct and outspoken veteran of the Xerox Palo Alto Research Center (PARC), who served as chief scientist at Atari before moving to Apple last year.

Kay now serves Apple as a sort of visionary and futurist. In the April 1984 issue of **St. Mac** magazine in an interview with Allen Munro, Kay expressed his view that in these initial stages of personal computer use, he himself is glad that some public schools are having trouble getting the funds to continue on their misdirected way.[1]

In this article/interview entitled "Alan Kay thinks the computer you just bought 'is no big deal'", Kay maintained that Silicon Valley engineers needed to read books and to quit building machines simply because they knew how to build them and simply because people would buy them. Now, Kay at Apple continues to look for **ways to make** computers facilitate human communication and interaction, education and social awareness. Encouraging people to teach Basic to six year olds at computer camps or in 11th grade "Computing" class is not what Kay is about.

Educators do not know if Seymour Pappert's conclusions hold valid assumptions or spurious speculation, but they do know that precious little concrete evidence indicates that little kids with computers are conceptualizing in remarkably astute and useful ways because of time spent in front of the Apple II-Plus. This does not imply that new cognitive breakthoughs by young students will not take place eventually as they compute with visionary new machines designed by people like Kay or work within educational frameworks suggested by scholars such as Peppert.

---

[1] Alan Munroe, "Alan Kay thinks the computer you just bought is 'no big deal.'" St. Mac, April, 1984.

What is known is that students like to feel important and capable, to act as if they are adults and can do adult things. The computer applications now available can help them to feel that way and to enhance their self-esteem, to make them feel a bit less uncertain about their abilities to get things done. "The ability to get things done" while admittedly simplistic, is often a usefully understandable definition of the word "power"- especially when you are 16 years old.

In United States public schools, there is as Theodore Sizer, has pointed **out in his marvelous study, Horace's Compromise: The Dilemma of the American High School**, a remarkable sameness to schools from coast to coast. Most have proms and homecomings, talent nights and track teams. Most have newspapers and almost all publish a school yearbook. In both of these co-curricular or extra-curricular activities, students assume a more than usual share of "real " responsiblity and depending upon their community and school leadership, a real share of power.

And, here is where technologically interested students and faculty advisers sense that they can do things faster and more effectively if the school will simply provide them access to the machines and software to so do.

* * * * *

James Madison Memorial Senior High School in Wisconsin's Capitol, Madison, is an upper-middle to lower-middle class Midwest American suburban high school. It's school newspaper, **The Sword and Shield** received the the 1984 Silver Crown Award from The Columbia Scholastic Press Association and received All-Columbian recognition in Content, Coverage, Layout, and Creativity. It's past editors have gone on to Harvard and Brown to Penn and Northwestern. Many have assumed "real world" responsibilities of some weight and merit after graduation.

Most recently, the newspaper which is entirely extra-curricular has moved toward computer production and these early moves have been documented in a segment of a 1985 Public Television production by WHA-Madison, the University of Wisconsin Television station. The series, **Computers in the Classroom** began local airing in January and will be nationally distributed this September.

Through the use of the Macintosh transportable computer, the staff and adviser have saved nearly $300.00 monthly in typesetting costs, and those savings have more than paid for the cost of the computer.

More importantly, the students have assumed greater responsiblity for typesetting, editing, billing, advertising design, and pre-production planning. In other words, the students have been permitted to act like journalists-in-training working at day-to-day journalism rather than treated as inept students who need coercion to do pointless academic exercise.

In addition, the adviser was able to use the computer in both English and Journalism classes to demonstrate certain basic concepts involved in the writing, editing, rewriting and presentation of written documents.

CLASSROOM OPERATIONS

What is significant is that the classroom and extra-curricular use of the computer was entirely devoted to practical applications of its power. The use of **Filevision** enabled the instructor to build and use a data base which students could use to pull up information on long- range planning projects and on their own progress or the progress of those working on a special group project. **Thinktank** enabled the instructor to design course outlines which could be collapsed into major headlines, easily modified as the course progressed as to better reflect actual classroom happenings, and edited into an accurate review sheet for students by term's end.

The word processing power of **Macwrite** permitted quick visual demonstration of principles of revision and modification which trained writers use regularily. Students were able to see how a good sentence could be made better by combining, eliminating, relocating or altering prases and word choices. They were able to appreciate the speed with which word processing could provide a clean, finished product.

A daily lesson would look something like the following: From the previous day's writing assignment the instructor would select five or ten already good, but slightly flawed or unpolished sentences. These would be placed on a handout and distributed to students. The instructor would then project one of the sentences onto a screen, use the copy function to duplicate and then proceed to demonstrate revsion skills drawing suggestions from students.

Quickly, students would see how an important idea took on more force if relocated at the end of the sentence or how a sentence sparkled when a more forceful verb was implanted. Sometimes the attempted revision wouldn't "work" and the result was the instructor forming a more flawed or more awkward construction. With good humored laughter, students expressed a new understanding of the basic fundamental of good rewriting: that the writer does indeed have options; that there are indeed alterative ways of expressing the same idea. They also grasped how laborious pencil-paper revision practices could be. More than a few would request personal computers for their graduation presents in June.

NEWSPAPER PRODUCTION

Five years ago, the production process from interviewing/ writing and revising news and feature articles for the school paper through the actual printng and distribution of the paper took two or three weeks of extra-curricular activity. Stories were frequently outdated by days before they finally appeared in print.

More importantly, from an educators point of view, entirely too much of the actual effort was done by professional copy readers, layout personnel and the printer himself. Initial galleys were typeset and returned after a few days to students who proofed and returned copy to the printer for corrections. Another exchange of copy saw the students do a pasteup which almost always contained gaps which the creative professionals then had to fill one way or another ie., by expanding a picture size or modifying a drawing. Frequent phone conversations were required between the faculty adviser or student editor and the print shop staff. When the monthly paper came out, it often looked very professionally done--which was too often the case.

Today, 90 per cent of every aspect of the newspaper is done or will shortly be done by students themselves. The faculty adviser will continue to provide logistical planning and long range continuity, advice on legal and libel issues, and friendly suggestions for improvement, but the computer with its word processing capabilities alone enables the students to write, revise, edit and print their own paste-up copy. At layout production time, students can reset single columns into double columns or change point size quickly. The **MacPaint** graphics program allows the business manager or graphics artist or merely an interested staffer to design an ad or pull up a clip art drawing **in minutes.** When the printer gets the pasteup sheets, he only has to half-tone the photos and make his negatives.

Significantly, each section editor can acquire skills and knowledge of broad aspects of a news business. In the past,the time involved required that effort be divided into specialized tasks. Now, a senior editor on the **Sword and Shield** should graduate having used Multiplan to check on advertising accounts receivable and bills payable, the status of the year's budget and the allocation of monies into paper or printing. He or she would likely have used a data base or program such as **Data Base II** or **Habadex** to add to or modify the mailing lists for newspaper exchange.

What are the prospects/implications for the future? It would seem likely that the practices of the newspaper staff will be emulated and improved on by others on the staff and in the classes at Memorial.

Already, the faculty adviser for Memorial's creative writing magazine, "Pressions" is having his staff prepare art, copy, and design plans using the Macintosh.

It would seem prudent for all school administrators to investigate the time and cost saving benefits which would result from the purchase of a laser printer which could then be used with a Macintosh to improve the final quality of the newspaper, creative writing journal, athletic and extra-curricular announcements, parent newsletters, course description booklets, program brochures, and theatre programs.

There is an interesting comparison to be made between this modest one-year experiment with classroom/ extracurricular use of the Macintosh and the impact of Visacalc on the sales of the Apple II.

Students this year liked the computer because it enabeled them to do specific practical tasks more easily, and not because they felt impelled to become "computer literate" in any narrow restrictive sense.

Businesspersons bought Apple II's so that they could run Visacalc and not because they felt compelled to become "computer literate"

Two groups of people simply had "things " to get done", and both the Macintosh now, and Visacalc then, helped them to get "things" done.

?! OTHER

"Clustering" In Vocational Business Education

Nathan Avani Ph.D.
State Supervisor Business/Office Education
Michigan Department of Education -VTES
P.O. Box 30009
Lansing, Michigan 48909

"Clustering" as a means for program organization and delivery of instruction in vocational business education is discussed in this article. Rationale for why we could cluster and its benefits for both students and employers is outlined. Occupational job titles and equipment recommendations are identified.

Technology, and its impact on all aspects of how we live, has been the theme of many articles and discussions during the past year. Education's role in preparing individuals for this technological boom and the microcomputer has been a challenge.

One instructional delivery system that reflects this technology and prepares individuals with employable skills is referred to as "clustering". Clustering, as defined by Donald Maley (1975): [1]

> "...is a form of vocational education which prepares the individual to enter into gainful employment in a number of occupations which have sufficient commonalities in human requirements and kinds of work to permit a high degree of mobility within, as well as job entry into, the occupations associated with the cluster."

Business and office is one such cluster area. This program includes the three occupational training areas of:

Accounting and Computing
Business Data Processing
Secretarial

In traditional vocational training, students would select one of these programs for occupational exploration and job preparation. This approach does not allow for the student to explore or acquire a job skill in either or both of the other programs that comprise the business and office cluster.

Clustering the three business and office programs into one provides the following for students:

(1) Opportunity for greater career awareness, exploration, and planning.

(2) Greater opportunity to match personal interests, abilities, and career goals with training goals.

(3) Greater opportunity to develop competencies identified as common to several job titles within the career area.

(4) More activities which infuse the content of two or more disciplines (accounting principles with programming;

data entry with letter production, etc.) thus giving the students a broader frame of reference from which to work and greater employability.

(5) Training in the organization and flow of work through a greater number of operations and job stations.

(6) Greater opportunity to develop operational proficiency on a variety of business/office equipment.

(7) Provide students with training in job titles currently offering employment opportunities.

(8) Provide composite career ladder and competency-based task listing which eliminates duplication of skill development between the three programs thereby allowing more efficient use of teacher time, equipment, and instructional materials.

(9) Greater opportunity for sharing and advisement through activities with representatives of advisory committees.

(10) A greater alignment between the training program and the current state of the art within the business community.

Students in the business and office cluster explore and acquire an employable skill in one or more of the following:

Accounting and Computing Occ. *

	D.O.T.
Inv. Control Clerk	222.387-026
Cost Clerk	216.382-034
Bookkeeper I	214.382-014
Bookkeeper II	210.382-018
Insurance Clerk II	205.567-010
Class. Control Clerk	210.382-030
Invoice Clerk	214.382-014
Payroll Clerk	215.482-010
Traffic Clerk	214.362-038
Gen. Ledger Bookkeeper	210.382-046
Accounting Clerk	216.482-010
Acct. Information Clerk	210.367-010
Credit Clerk	205.367-022
Billing Control Clerk	214.387-010
Audit Clerk	210.382-010

Data Processing Occupations

Data Entry Operator	203.582-030
Terminal Operator	203.582-054
Computer Programmer	020.162-014
Computer Operator	213.362-010

Secretarial Occupations

Adm. Secretary	169.167-014
Word Processing Correspondence Secretary	203.362-500
Stenographer (Comm. Specialist)	201.362-030
Transcriptionist	202.362-014
Clerk/Typist	203.362-010
Typist	203.582-066
Correspondence Clerk	209.262-010
File Clerk (Information Specialist)	206.362-010& 206.367-014
Reprographics Clerk	207.682-010& 207.682-014& 207.682-018& 207.685-014
General Office Clerk	209.562-010
Mailing and Order Clerk	209.587-026& 249.367-038
Shipping, Receiving and Stock Clerk	219.367-030& 219.367-034
Proofreader	209.387-030
Traffic, Rate, and Transportation Clerk	214.587-014& 214.362-038
Medical Transcriptionist	202.362-014
Medical Secretary	201.362-014
Medical Assistant	079.367-010
Legal Secretary	201.362-010
Receptionist and Communications Systems Operator	207.367-038

*D.O.T. (Dictionary of Occupational Titles, Fourth Edition, 1977).

The cluster program is housed in one facility. The physical surroundings and arrangement of the cluster area can be as important as the type of equipment to be located in the facility. All students and staff should have free and easy access to both equipment and instructional materials. It is also important to have as few visual obstructions as possible (walls, dividers, etc.) in the cluster area to allow for the staff to be able to visually observe all students and for equipment layout efficiency. It is beneficial to have available one or two areas adjoining the main cluster area for delivery of lectures, dictation, small group sessions, etc. This area would consist of tables and chairs, blackboard area, screen, overhead projector, and other typical classroom materials. The cluster area should be arranged so that each student has his/her own work area comparable to what business and industry would provide. The equipment list below is based on a ratio of 66:3, that is 66 students to 3 teachers.

Data Processing Equipment

Qty.	Item
12	CRTs (display terminals)
1	Central Processing Unit (CPU)
	Input/output peripherals:
	(a) High-speed printer
	(b) Fixed disk drive
	(c) Removable disk or type unit for backup
	(d) Diskette reader or CRTs for input
1	High-speed printer (on-site)

Data Entry Equipment

6	Data Entry Devices (one or more off-line)
	(a) Key-disk units
	(b) Key-tape units
	(c) Off-line entry key diskette units

Microcomputer Equipment

22	Microcomputers
6	Printers:
	(a) Daisy wheel/letter quality
	(b) Matrix

Word Processing Equipment

4	Word Processors:
	(a) Dedicated
	(b) Computer based
1	Letter-quality printer
	Calculators (combination of):
33	(a) Electronic printing
	(b) Electronic display
	(c) Electronic display/printing
22	Typewriters
1	Telephone system (at least 6 phones)

Copying Equipment

1	Good quality copying machine

Dictation/Transcription Equipment

11	Dictation/Transcription Units
66	Workstations
66	Chairs for workstations
11	Carrels and chairs
22	Slide/cassette machines
22	Tape recorders/headphones
1	Paper cutter
15	Storage cabinets
15	Vertical/lateral files

Students in the cluster begin by going through a comprehensive career development component that expands their awareness and understanding as to the various occupations available for training. The students then go through a series of activities designed to assist them in selecting appropriate occupations for training. At this time, an individual student career plan is developed and activities take place for job skill development.

Along with skill development activities, each student is involved in completing "core competencies" that would be applicable to all, regardless of training selections.

As equipment changes take place, so must our delivery system for occupational training. If we are going to meet the challenge of technology and the implications it has for instructional delivery systems, we need to "step out" and take a serious look at the way we deliver instruction. Hopefully, the "cluster" concept is one way we can reach this goal.

[1] Maley, Donald, Cluster Concept in Vocational Education, American Technical Society, 1975.

Tips For Teachers On Organizing Data

Muriel Fung
The Ontario Institute for Studies in Education
252 Bloor St. West
Toronto, Ontario M5S 1V6

This paper examines a number of current methods in organizing and managing educational data utilizing the computer as an analytical instrument. Methods of statistical inquiry are discussed. Item analysis, computer programs, and ways to improve communication are included.

Introduction

Most classroom teachers have valuable data on their students. Records are kept of their students' academic achievement, health, personal development, social skills, and class attendance. Management of the data is an overwhelming task. It is so tedious and time consuming that most teachers would rather leave it alone until the report cards are due. Very often, great effort has been spent on tabulating such data, yet rows and columns of figures on a sheet of paper may convey little or nothing to the readers; namely, parents, the principal, or even the teacher. Teachers need to process these figures into interpretable and useful information. Microcomputers have the capability of helping teachers to organize their data.

Computers in the schools are mostly used as teaching and learning tools. A variety of functions are: drill, simulation, information retrieval, and problem solving. Yet very few teachers utilize the computer to keep track of the students' progress; e.g. as a form of record-keeping, statistical analyses and the like.

Most teachers have informal skills on processing data. They make day-to-day assumptions and decisions or statistical statements even though they may not have performed calculations. Their teaching experience provides a base to make predictions for a particular student's achievement. In evaluation of CAI packages, often by intuition, they are able to discriminate between the good and bad. But in this information age when more accurate and interpretable information for decision making is required (Molnar, 1979), it is important that these skills be formalized and the data be treated more scientifically. This paper addresses this new issue. It will focus on methods of organizing data utilizing the computer as an analytical instrument. Methods of statistical inquiry as well as some software packages will be discussed. Ways to improve communication will also be dealt with.

Methods of Statistical Inquiry

Teachers observe their students, compare test results, then look for connections and patterns, similarities and differences which they happen to have noticed. They make some conclusions as to what might have happened. The tendency to generalize is an essential part of their thinking. They want to be able to generalize with confidence. This is where statistics comes in. Statistics helps teachers to look for reliable information in the classroom. It enables them to summarize and predict more precisely a variety of situations involving measurements or observations of quantitative material. Knowledge of statistics is important in that it provides tools which lead to a clearer understanding of educational data.

Suppose that the principal of the school asks how the class did on a mathematics test.

Instead of listing all the students and their scores, the teacher could give the average score for the group. This involves descriptive statistics which helps to transform a large set of data into a more interpretable form.

By using methods of statistics (Mendenhall, 1975), (Harnett, 1975), e.g. frequency distributions, the teachers can see the most frequently occurring scores. They can also identify any pattern of distribution; whether it is normal or skewed (Bohrnstedt, 1982). They might be interested in obtaining a measure of the central tendency which gives a numeric index of the average score of the distribution. The two most important measures of the data are the arithmetic mean and standard deviation (Ferguson, 1976). When data on two classes are available, teachers can compare the means and the standard deviations to show differences. By using the analysis of variance method, several groups of the same sample can be compared for significant differences. Finally, teachers might like to know the relationship between a pair of variables, e.g. Reading and Math scores. Correlation is a measure of this relationship and the coefficient indicates its strength.

It is important to know that qualitative data can be processed analytically (Miles, 1984). Information derived from classroom observations and data obtained through a regular log or diary in narrative text form can be organized in a systematic way that final conclusions on students, events or processes can be drawn and verified. Further, there are scaling procedures (Nishisato, 1980) which enable the teachers to assign numerical values to alternative or categories so as to discriminate optimally among objects. The DUAL3 (DUAL3,1984) software series can perform these functions. The present trend is to encourage educators to develop better qualitative methods.

The methods described above may be oversimplified. The author wants to convey the message that the application of statistics to education is not as mysterious as it sounds. Statistics is only a means to summarize the information collected, so that it can be more readily comprehended. The teacher's task is to reduce the massive data to some summary measure by statistical analysis while it still presents the essential information of the classroom data.

Graphic methods

A number of graphical methods, bar charts, histograms, pie charts, scatterplots and other pictorial techniques, can be used to present data. Illustrations of this nature can give the dimension of various factors in their perspective so that the teachers may discern differences in shapes, patterns and other physical magnitudes (Chambers, 1983). Graphic representation of data can be extremely valuable for communicating information quickly and clearly to the viewer. Suppose a question is asked whether there is a relationship between two sets of marks; a scatterplot may be used to demonstrate the relationship or lack of it. A number of microcomputer graphic packages (Chartmaster), graphic hardware, terminals and pen plotters are widely available at a reasonable cost.

Test Scoring and Item Analysis

Preparing multiple choice questions and marking papers are tedious activities. There are computer programs (Koprin, 1984) to generate items and to perform clerical tasks. Some packages can even quiz the students, evaluate their responses, make a testing decision and provide immediate feedback. By printing a profile of the item scores, the strengths and weaknesses of the students are more accurately presented. In short, item analysis also can point out deficiencies (Granlund, 1977) in the questions and can provide teachers with data they always meant to collect but were never able to find time to do so. By using these procedures, teacher-made tests can be constructed to realize teaching-learning objectives. Improving the effectiveness of classroom tests would consequently improve the nature of the education process itself.

Some Useful Computer Programs

With the advent of high-power microcomputers to the markets, sophisticated statistical packages are available. Statistical programs such as SPSS (Nie, 1975) and MINITAB (Ryan, 1976) written for large computers are now available for microcomputers. These programs may not have the full capability of the main frame computers, yet similar routines for a smaller data set can be processed.

There are also a number of other statistical programs developed for microcomputers. Some simple programs can perform descriptive data analysis (P.C. Statistician), (Statistics Series) while more sophisticated ones can do complex factor analysis (Principal Component and Factor Analysis) or multiple regressions (Multiple Regression). Most of these programs are flexible and user-friendly. It is generally known that even a perfectly formulated statistical program would not guarantee accuate results. As numerical precision is crucial to any statistical program, the merits and limitations of the software should be evaluated before it is purchased.

The electronic spreedsheet (VisiCalc) is a versatile instrument for the teachers. It allows teachers to enter text, data and formulas, then command the program to perform calculations. Some spreadsheets, (Lotus 1-2-3), can be integrated with graphic programs which makes it possible to enter raw data, perform the necessary calculation, then incorporate the data into a graphic output through a plotter.

Communication is an important part of teachers' skills. Word processing through the computer enables teachers to produce a final professional-looking copy in less time than making a draft report. There is also the spelling checker (Wordstar & Spellstar), thesaurus program and mailing list generator (Mailstar) for creating form letters.

Conclusion

This paper examines a number of current methods in presenting and analyzing educational data. It is becoming increasingly clear that microcomputers have played an important role in this process. The impact of the information age calls for more precise and scientific methods of fact finding. Teachers must recognize the dynamics of information processing technology and the role they can play to enhance analytical understanding. It is appropriate that they should acquire the skills and concepts of scientific organizing and managing data to function effectively in an information society.

REFERENCES

Bohrnstedt, G. et al "Statistics for Social Data Analysis", Itasca, Ill. F.E. Peacock Publisher, 1982

Chambers, J.M. "Graphical Methods for Data Analysis", Duxbury Press, Boston, 1983

Chart-Master, Customer Service Plan, Decision Resources Software Designed for Decision Makers. Westport,CT.

Ferguson, G.A. "Statistical Analysis in Psychology & Education," McGraw-Hill, N.Y., 4th Ed. 1976

Granlund, N.E. "Constructing Achievemnet Tests", Prentice-Hall, 1977

Harnett, D.L. "Introduction to Statistical Methods", Addison-Wesley Publishing Co., 2nd Ed. 1975

Koprin, L. "Computer Assisted Testing Operations", Computers in Education, Sept. 1984, pp20-22

LOTUS 1-2-3, Lotus Development Corporation, Cambridge, MA

Mailstar, Micropro Softwares, San Rafael, CA.

Mendenhall, W. "Introduction to Probability and Statistics", Duxbury Press,MA, 4th Ed. 1975

Miles, M.B.,& Huberman, A.M. "Qualitative Data Analysis: A Source-book of New Methods", Sage, CA. 1984

Molnar, A.R. "National Policy Toward Techological Innovation and Academic Computing", The Journal, Vol.4(6), Sept/Oct, 1977

Nie, N.H, et al, "Statistical package for the Social Sciences", McGraw-Hill Book Co., 1975

Nishisato, S. "Analysis of Categorical Data: Dual scaling and its Applications", U. of Toronto, Toronto, 1980

P.C. Statistician, Human Sysytems Dynamics Inc. Northridge, CA.

Principal Component and Factor Analysis, Dynacomp Inc., Rochester N.Y.

Ryan, T.A. et al, "MINITAB Student Handbook", Duxbury Press, MA, 1976

The DUAL3 Statistical Software Series, MicroStats, Toronto, Canada, 1984

The Statistics Series: Stats Plus, Human System Dynamics Inc. Northridge, CA.

VisiCalc, VisiCorp, San Jose, CA.

Wordstar and Spellstar, Micropro Softwares, San Rafael, CA.

Research Needs for Computers in Education

Ronald G. Ragsdale
The Ontario Institute for Studies in Education
Toronto, Ontario, Canada

Computer applications in education have moved forward rapidly on the basis of intuition, rational analysis, manufacturer pressure, luck, and hard work. A solid research base is needed in order to consolidate the gains and pave the way for further advances. This paper includes an analysis of the basic research needs, plus a more detailed description of some areas requiring special attention.

INTRODUCTION

Although the activity in educational computers applications is impressive and increasing, the amount of activity may not be an accurate indicator of the progress being made. Specifically, sharing of knowledge about the effective use of computers in education, both historically and currently, seems to be messing. For example, the experiences which were a part of mainframe applications in the 1960's were repeated on minicomputers in the 1970's and are now a part of the microcomputer age. This repetition is demonstrated in the reprinting of two 1960's works by Patrick Suppes in *The Computer in the School: Tutor, Tool, Tutee* (Taylor, 1980).

The lack of communication has resulted in some duplication, which is not always bad, since replication can either lead to confirmation of results, or even more interesting, unexpected differences. Unexpected differences are also called side effects, the detection of which is necessary to any evaluation of computers in education.

Side effects are likely to be an important source of information about the impact of computers in education because computers are a communications medium. The history of communications media has shown that the true effects of media are usually not determined until long after they have been implemented. Similarly, the implementation of a project may be aimed at different objectives than were originally intended. One cause of this lack of consistency is that many people may agree on general goals, while disagreeing on specifics. As implementations become more specific, the possible outcomes may move away from the original goals.

Because computers (as communications media) are producing unpredictable results, while the projects we implement may not be directed toward our original goals, a *continuous* monitoring, or needs assessment is required, with particular emphasis on the detection of side effects.

The implications for research lead to determining current needs, assessing the effects of computers on the educational process, finding the new needs which these effects create, and feeding these results into a continuing process.

The following sections describe the components of such a needs assessment process, including assumptions, curriculum orientation, and iterative nature of the process, plus the role of values, the communications effect, and intentionality in the process. Also discussed are the possible sources of distorting inputs to an educational needs assessment.

NEEDS ASSESSMENT

Needs assessment should be a continuous process whereby the requirements of the curriculum, students, teachers, and parents are considered as fundamental to making any educational change. In the case of implementing computer innovations in classrooms, a needs assessment would emphasize human requirements and not the attributes of the technology. The tendency to expect children and teachers to adapt to the demands of computers reverses the role that technology should play.

The process of needs assessment is essential to effective planning, but must be conducted accurately and frequently to guide the use of computer applications in education.

Assumptions

Due to the evolving nature of computer characteristics, needs assessment is a continuous process. However, the focus of the process must be on the curriculum (future, as well as current), not on the attributes of the technology. It may be that experience with computers will be more of a hindrance in determining these needs, since the experience will tend to bias our expectations. Therefore, the primary source of input for a needs assessment of writing training would not only come from those who teach writing, but would include (or perhaps emphasize) those who do not use computers.

The outcome of the process should be in the form of the functions which need to be performed or, in the case of computers, software characteristics, based on curriculum goals, independent (as much as possible) of hardware attributes. (It should be noted that, in practice, this is very difficult to do. Hardware requirements must usually be set well in advance of the creation of software because the software must have hardware on which to be created. This can be a difficult timing problem, since it is important to work with advanced hardware to develop new techniques, but one must not get too far ahead of the potential users.)

Curriculum orientation

One difficulty with the curriculum focus is that, in addition to what we do now, it must also include what we would like to do. For example, the goal of our writing instruction might be set by determining the techniques which professional writers use to make their writing more effective, even though some of these techniques are beyond the capabilities of our present students.

It is almost impossible to focus on the curriculum without any knowledge of computer potential, but worth trying. This might be accomplished by questioning those who have no computer experience as well as asking those who have used computers to put their experience aside. This phase might be followed with one which explicitly considers computer characteristics, an approach which is similar in some ways to Scriven's "goal-free" evaluation (Scriven, 1974).

Iterative

Any identified needs can only be tentative. Alterations will occur because of changing curriculum goals, technology advances, and the results of previous need-assessing cycles. For example, it might be found that the use of word processing has resulted in some negative effects, such as the indiscriminate re-cycling of previously used text (leading to coherence problems). One of the most important forces in changing the needs will be the changing applications of the new technology. An example is word processing, which wouldn't have been an identified need only a few years ago, is now seen as a strong need, and whose use brings other needs into focus.

Once the initial needs are determined and strategies to satisfy them are implemented, the needs assessment process would be repeated in order to detect the new needs which would be surfacing. At this point, it probably bears repeating that the needs assessment has many inputs which are subjective and the entire process can be no better than the values which underlie it.

THE ROLE OF VALUES

One must be aware that the needs assessment process itself is strongly influenced by the values which are fed into it, both through the goals of the educational system and the values of those who conduct the assessment (Guba & Lincoln, 1982). For a detailed analysis of the effect of scientists' and engineers' values on the history of technology, see Noble (1984). The features of computers can impose their own values through their functioning as communications media and the deficiencies of our intentionality.

Communications Media

The use of computers as a communications medium is directly connected to the word processing applications, but also has some implications for our ability to predict their ultimate use and their importance in our lives. Edward Lias, in his book *Future Mind*, described some of the principles of media, emphasizing the difficulty that people have had in predicting the ultimate uses of these media (Lias, 1982). To illustrate this point, he described an incident from 1876, when the then President of the United States, Rutherford B. Hayes, inagurated a new long distance phone line between Washington and Philadelphia. Hayes was unfamiliar with telephones and ill at ease as he carried on a short conversation to officially open the service. After performing his duties, he stepped aside and said to an onlooker, "It's a very interesting invention, but why would anyone want to have one of them?"

At about the same time, a British newspaper carried an article which commented on the increasing use of telephones in North America, but pointed out that they would never be that important in Britain, because "we have such an ample supply of messenger boys". In both cases, the emphasis was on the way things were currently being done, with no anticipation as to how the telephone might change their ways of doing things. In considering word processing and other computer applications in schools, we must be aware of how easy it is to make this mistake.

Since the future applications of computers are so difficult to anticipate and predict, we must be extremely flexible in our planning, allowing for continual modification. In addition, we will need to constantly seek input from computer users, in order to keep up to date on the changing applications of computers. For example, teachers of writing and their students are important sources of input, along with experienced writers and others using word processing.

Intentionality

Needs assessment can be seen as a statement of intentionality. In the area of computer technology, as in the U.S. space program and mountain climbing, the desire to do something "because it is there" is extremely strong. However, agreement on what is to be done is, in itself, often not sufficient to move those who agree toward their seemingly common goal. That is, what is called the "rational systems" approach is often not adequate for the task of accomplishing the intended goals (Seidman, 1983).

Seidman describes the rational systems approach to educational change as beginning with legislation or judicial decisions, then proceeding through the development of program guidelines, implementation by local school authorities (driven by the rewards or sanctions in the guidelines), and including evaluation to determine compliance with the guidelines, as well as the quality of the program. Unfortunately, after the first step in which legislators or jurists agree on vaguely stated general goals, a series of compromises in the other steps may result in outcomes that are the opposite of what was intended. That is, what can be defined in a guideline, implemented in a school board, or evaluated quantitatively is often different, in important ways, from what was originally intended by those who set the process in motion.

Changes in the Educational System

It seems clear that the selection of appropriate technology for education should be based on the needs of the students, teachers, and parents. However, many of these needs, at least as they intersect with the potential benefits of technology, have not been completely determined. The result is the selection of technology in a manner unrelated to the real needs of people in the educational system.

When the needs of the people, particularly students, are not clearly identified, or are ignored, other sources of needs may replace them. The most dangerous are the needs of technology itself. By this is meant the implementation which focuses on the attributes of the technology and attempts to maximize the use of unique characteristics. It is equivalent to asking, "What can computers (or some other technology) do best?" Much planning and prediction in education seems to be based on the assumption that what can be done should be done.

A second source of potential needs is at the management level. The needs which might arise include those of replacing teachers, or making teachers conform to procedures set by administrators. The danger is that needs which ease the life of management may be seen as more important than student learning needs.

Although teachers might be targets of management needs, they can also be a source of competing needs. Technology which is misused, or under-utilized, such as overhead projectors or programmed instruction, can be thwarted by the people who must use it. The needs of teachers may obscure the potential benefits of technology.

Finally, the students themselves may be a source of competition to their own needs, if their *wants* are given greater weight than their *needs*. An example can be seen in the history of television. Basing choices of technology on student wants might lead to an instructional wasteland to rival the wasteland of television.

CONCLUSION

A focus on the needs of the educational system does not lead to consensus without a common basis for the source of those needs. Agreement on broad general goals does not lead to consensus on required actions. Finally, agreement on the actions required (intentionality) may still not be sufficient to bring about the desired results.

The aims of a research program should be careful and ongoing assessing of educational system needs, selecting and monitoring proposed actions to meet these needs, and evaluating the impact of these actions, including both intended outcomes and side effects.

REFERENCES

Clark, R.E. Reconsidering research on learning from media. *Review of Educational Research*, 1983, *53*(4), 445-459.

Guba, E.G., & Lincoln, Y.S. The place of values in needs assessment. *Educational Evaluation and Policy Analysis*, 1982, *4*(3), 311-320.

Lias, E.J. *Future Mind.* Boston: Little, Brown, 1982.

Noble, D.F. *Forces of Production: A Social History of Industrial Automation.* New York: Alfred A. Knopf, 1984.

Scriven, M. Prose and cons about goal-free evaluation. In W.J. Popham (Ed.) *Evaluation in Education*, Berkeley, CA: McCutchan, 1974, 34-67.

Seidman, W.H. Goal ambiguity and organizational decoupling: The failure of "rational systems" program implementation. *Educational Evaluation and Policy Analysis*, 1983, *5*(4), 399-413.

Taylor, R.P. (Ed.) *The Computer in the School: Tutor, Tool, Tutee.* New York: Teachers College Press, 1980.

Constructing Computer Interfaces For The Laboratory

Bill Rouse
Miami University
Oxford, OH 45056
USA

Microcomputer Based Instrumentation constitutes one of the major areas of computer application in our society, yet our educational systems largely ignore it. School systems must acknowledge MBI as a legitimate part of computer literacy, especially for students preparing for careers in science and technology. Teachers can utilize MBI packages specifically designed for instructional activities or adapt general purpose MBI interface boards for use in the classroom, but the author advocates the practice of using teacher-constructed interfaces and suggests some means to assist the teacher in learning this technolgy.

Since the beginning, human beings have found many ways to use computers, and with decreasing costs, miniaturization, distributed processing, and increasing availability, the variety of computer applications has increased. And as a general rule, each time society has found a new application for computers, teachers have adapted it to the realm of education. An exception to this rule is microcomputer based instrumentaion.

Microcomputer based instrumentation (MBI) is the use of microcomputers to directly measure, monitor, or control physical phenomena or devices. Such use requires a hardware/software interface between the physical world and the microcomputer.

Consider an example of measurement. A thermistor is used as a temperature transducer to create variations in an electrical signal that are proportional to the variations in the temperature of the substance being measured. A signal conditioning circuit may be required to amplify or transform this signal so as to obtain a voltage that varies from 0 to 5 volts in proportion to variations across some temperature range such as 0 to 100 degrees Celsius.

This voltage is an analog signal, i.e., a signal that varies continuously from one value to another as opposed to a digital signal which is represented by discrete values and which the computer can process. So a circuit called an ADC (analog to digital converter) is required to convert the varying voltage into a corresponding digital signal. If the ADC has an 8-bit output it converts the analog variations of 0 to 5 volts into digital variations of 0 to 255.

By including an algebraic conversion in the system software, the 0 to 255 scale can be displayed on the video screen as a 0 to 100 degree scale. Therefore, a temperature of 100 at the transducer will generate a corresponding 5 volt analog signal which will be converted into a corresponding 255 digital signal which will be displayed on the screen as 100 degrees. Other temperatures will result in proportional analog, digital, and video signals, e.g., a temperature of 50 degrees will generate a 2.5 volt analog signal, a digital signal of 127, and a video signal of 50.

We now have an example of an overpriced thermometer! And it would be overpriced if this were all it could do. However, software can cause the system to take temperature samples from the substances being used in an experiment every five minutes, night and day, over a period of a week, and store them on a disk or graph them on a printer. By adding some relays the system can be modified to make decisions based upon temperature to activate or deactivate heating or cooling units. Now the example is one of monitoring and controlling.

The same ADC used in the example above could be used with other transducers to monitor other phenomena. Motion can be detected by using a light sensitive transducer that will cause the ADC to report one number when light strikes the transducer and another number when the light is blocked. By continually monitoring these numbers as they are reported, the computer can determine when an object passes the transducer and blocks the light. Thus, the average velocity of an object can be determined by using a pair of transducers located a given distance apart and counting the elapsed time for an object to move from one to the other. If 8 transducers are positioned at regular intervals along an inclined plane or air track, so that the car passes 7 pairs of transducers, then as the car descends the incline, the system can calculate its average velocity at 7 locations and its average acceleration at 6, and display these values on the video screen.

These examples suggest only a few of the applications of MBI. Research labs use MBI for monitoring experiments and for direct data gathering. MBI controls medical equipment. MBI is essential to robotics. Factories use MBI to control assembly lines. New industries are springing up that produce MBI systems for old industries. Even our cars use MBI to control ignition systems, to predict when we will run out of fuel, and to chastise us for not buckling up. MBI permeates our society.

But we seem to be keeping it a secret from our students. Our school systems almost completely ignore the existence of MBI. Our computer education experts define and advocate a computer literacy that appears to be completely devoid of any mention of MBI. How can a student be computer literate without knowledge of microcomputer based instrumentation? We must insure that at least students pursuing technological or scientific studies use MBI and know how it works.

Science and mathematics are two areas in which it is appropriate for students to use MBI. Science is an obvious area because of its concern for measurement and control of physical phenomena and devices. Mathematics may not be so obvious but MBI is rich in opportunity for application of mathematics to real world problems such as those arising from sampling and analysis of data, mathematical modeling, and calibration of measuring devices. Teachers should give students opportunities to use MBI for data gathering in order that they have experience with techniques and equipment similar to those being used in the real world.

Science and computer science are two areas in which it is appropriate for students to learn how MBI works. The techniques of MBI interfacing is a legitimate topic for a science class both in terms of electronics content and as a capability for future use in school and on the job. And because of its relation to computer architecture, interfacing is surely appropriate to the computer science curricula that are emerging in many school systems today.

For a teacher who wishes to incorporate MBI into the curriculum two routes can be taken: purchasing ready-made devices or making her own.

Many companies produce MBI interface boards that can be adapted to school use. (See list at end of paper.) These boards are generally of high quality, but the buyer must supply transducers and supporting devices and write the control software to make them work.

There are a few companies that market complete packages including interface boards, transducers, software, and even documentation for school applications. (See list at end of paper.) Packages have the advantages of being complete and useable and designed expressly for instructional purposes. They have the disadvantages of being designed for specific tasks and may not be easily adpatable to other tasks or modes of operation, being relatively expensive especially when several sets are needed to set up multiple stations in a lab, and being prefabricated so they do not give the teacher nor the students the depth of insight into the technology that they would get if they fabricated the systems themselves.

Now consider the option of teacher-made systems. The main disadvantage of this approach is that the teacher will not immediately have an operable system on line, since the teacher will have to construct the hardware and write the software to control it and prepare the notes for students to use during the lab activity. However, the advantages outweigh the disadvantages. Electronic parts for hardware construction are much cheaper than ready-made boards or systems, so that more systems can be acquired for the same ammount of money. The teacher-made system can be designed to more closely fit specific purposes. Teachers and students who build their own systems know how their systems work and get insights that help to demystify the technology.

But for many teachers, even some with electronics backgrounds, the idea of constructing an electronic interface is scary! They think electronics is a complex subject. Well, they're right! However, much of that complexity is now being packaged inside electronic chips (integrated circuits) and a person need know little more than how to connect them together in order to accomplish a desired task. An extensive knowledge of electronics is not required in order to do some significant and exciting interfacing for the laboratory. For the past three years I have conducted interfacing workshops for elementary school, secondary school, and college teachers. Many of these teachers had completed only one year of physical science. It has been my experience that the most important prerequisites for learning to interface computers is a basic understanding of AC and DC voltage, current, and resistance, and liking to build things with ones hands. A person with these prerequisites is ready to study MBI interfacing.

Although it is possible with motivation and perseverance to develop a thorough working knowledge of MBI through self study, the most efficient way to begin is to take part in a hands on workshop. Such workshops are conducted for people in industry, but they tend to be quite expensive and oriented to industrial applications rather than instruction. Only a few persons conduct interfacing workshops specifically for teachers. (See the list of addresses at the end of the paper.)

Robert Tinker of the Massachusetts Institute of Technology has pioneered the educational uses of MBI and has established a private corporation, the Technical Education Research Center, that offers workshops on a variety of microcomputer topics including MBI at various locations around the United States. The Center also publishes a newsletter called Hands On!.

I am not aware of his current plans, but in the past Eric Salin of McGill University has offered short courses on interfacing for the laboratory.

Linda Leffel is the contact person for computer interfacing workshops offered by Virginia Polytechnic Institute.

As mentioned above, I regularly offer workshops on computer interfacing for teachers. These are two-week, credit granting workshops offered through Miami University and scheduled for late spring or summer.

Although only available to college instructors in the United States, the National Science Foundation usually includes short courses on computer interfacing in its Chautauqua program.

The Event Queue, a regular feature of Byte Magazine, is a good place to look for announcements of such workshops.

For self study, either separately or to supplement the knowledge gained through a workshop, books on interfacing are a good resource. Many books have been written on the subject, some dealing with interfacing in general and some dealing with specific microcomputers. I have yet to find a comprehensive text that deals effectively with MBI for the beginner, but many contain much useful information. A search through the book racks of a local microcomputer dealer should turn up some useful resources.

Certain journals are good resources for self study. (See list at end of paper.)

Hands On! has been mentioned above.

The Circuit Cellar, written by Steve Ciarcia, is a regular feature of Byte Magazine and consists of articles on construction projects of various types.

The Computer Journal ("for those who interface, build, and apply micros") is a very small publication but one that contains much useful information.

Finally, a valuable resource that most people would overlook are hams, i.e., amateur radio operators. As a rule hams are knowledgeable about state of the art electronics, have a propensity for public service, and are delighted to share their knowhow. Amateur radio has a long history of innovation in electronics and many hams are currently interfacing their rigs to computers. Make friends with a ham and ask for help. (My call is N8ADD.)

Properly interfaced microcomputers make powerful tools for demonstrations and experiments in the classroom and the laboratory. They allow students to use up-to-date methods of data gathering and to concentrate their attention on the physical principles being studied. They let students gain experience with computer control applications; not merely with electronic page turning. Teachers who employ MBI give students experience with the state of the art techniques used in the outside world, rather than with the outmoded practices no longer used except in school.

WORKSHOPS

Bill Rouse
301 McGuffey Hall, Miami University
Oxford OH 45056 USA.

Eric Salin
Chemistry Dept., McGill University
Montreal, Quebec H3A 2K6 Canada.

Linda Leffel
C.E.C., Virginia Tech
Blacksburg VA 24061 USA.

Robert Tinker
Technical Education Research Centers
8 Eliot St., Cambridge MA 02138 USA.

W.R.Zeitler, NSF Chautauqua Coordinator
Science Education, Univ. of Georgia
Athens GA 30602 USA

JOURNALS

Byte Magazine
POB 590
Martinsville NJ 08836 USA

Hands On!
8 Eliot Street
Cambridge MA 02138 USA

The Computer Journal
POB 1697
Kalispell MT 59903 USA

INTERFACE BOARD MANUFACTURERS

AST Research
2121 Alton Avenue
Irvine CA 92714 USA

California Computer Systems
250 Caribbean Drive
Sunnyvale CA 94086 USA

John Bell Engineering
1014 Center Street
San Carlos CA 94070 USA

Mountain Computer
300 El Pueblo Road
Scotts Valley CA 95066 USA

SSM Microcomputer Products
2190 Paragon Drive
San Jose CA 95131 USA

EDUCATIONAL PACKAGES

Cambridge Development Laboratory
100 Fifth Avenue
Waltham MA 02154 USA

Interactive Microware
POB 139
State College PA 16804-0139 USA

Computer Attitudes And Knowledge Among Students And Educators

Jon C. Marshall
N.C. State University
ELPE Dept., Box 7801
Raleigh, NC 27695-7801

Susan H. Bannon
University of Missouri
AFSE Dept.
St. Louis, MO 63121

The study was designed to determine the levels of computer knowledge and computer attitudes among secondary students, university students, and educators, and the relations between these variables and selected demographic characteristics. More than 1900 students, 400 teachers, 70 school administrators, and 80 media specialists were included in the sample. Among the findings were that students more frequently than educators had access to home computers; there is no relation between having access to a home computer and computer knowledge or computer attitudes; and teachers know more about computers than do their students.

Introduction

Confronting educators at all levels are policy decisions concerning microcomputer utilization. Questions like:

- Who are the computers for?
    - Teachers?
    - Administrators?
    - Students?
- How many computers do we need?
    - One in every building?
    - One in every classroom?
    - Multiple computers in every classroom?
- What computer should we purchase?
    - TRS80?
    - Apple?
    - Atari?
    - IBM?
- What software is needed?
- How can the teachers, administrators, and students be taught to use the computers?

This list of questions could continue almost indefinitely. And, they reflect important decisions which are going to affect educational policy for both the short and long term.

As important as these decisions are to the future of education, they frequently are based on general impressions and, at best, partial information. An adequate data base has not been available to decision-makers. There has been a wealth of often confusing and contradictory technical data available. A lot of assumptions have been made about the attitudes and knowledge pertaining to computers of teachers, administrators and students. But, little actual data have been available.

In general, the research reported over the last decade suggests that students and adults have positive attitudes toward computers. For teachers these attitudes seem to be less favorable when faced with computers in their own classrooms. There is little comparable information on computer knowledge levels or how knowledge interacts with attitudes.

Problem

The purpose of this study was to determine the computer attitude and knowledge levels for students and teachers, the interaction between attitude and knowledge, and the relations between these two variables and selected demographic characteristics.

Procedure

The general method for this study was to administer a computer survey to students in grades nine through university and educators employed in grades kindergarten through university. All subjects were from Central and Eastern Missouri and Southern Illinois. The breakdown for the sample is:

- Teachers - 423
    - Elementary - 161
    - Secondary - 223
    - Other - 39
- Students - 1811
    - Secondary -1210
    - University - 601
- Administrators - 72
- Others - 169

TOTAL - 2475

The survey consisted of the following four sections:

1. Attitudes about computers - 18 items (These items were rated using a five-point scale ranging from Strongly Agree to Strongly Disagree)

2. Computer Knowledge - 15 items (These items were arranged in 3 levels of 5 items each--basic literacy, intermediate, and advanced.)

3. Demographic - 4 items (Sex, Age, Highest Degree and Availability of home computer.)

4. Specific Questions for Subsamples (Students-3 items, Teachers-6 items, Administrators-7 items, and Library/Media Specialists-4 items.)

Responses to the survey were collected using NCS General Purpose answer sheets. Data analyses were completed using SAS routines.

Results

Both descriptive and comparative data will be presented at the ECOO/AEDS Conference. This paper will focus on the development of the attitude scales and the interrelations between these scales, computer knowledge and the demographic variables.

Two attitude scales were identified from the 18 items on the survey. These scales are:

1. Cognitive Computer Attitudes
    - Mean = 8.024
    - Standard Deviation = 3.776
    - Alpha Reliability = 0.929
    - Number Items = 7
2. Affective Computer Attitudes
    - Mean = 17.055
    - Standard Deviation = 4.155
    - Alpha Reliability = 0.896
    - Number Items = 7

Scoring for the items in each scale ranged from 0 (Strongly Agree) to 4 (Strongly Disagree). The possible score range was 0 to 28. The mean of 8.024 for the first scale suggests that the students and educators had generally positive cognitive attitudes; the mean of 17.055 on the second scale reflects a tendency toward rejection of negative affective attitudes.

The mean computer knowledge score, was slightly over 5 points out of the 15 points. This suggested that on the average students and teachers had a basic literacy knowledge level about computers.

The interrelations were determined between computer knowledge, cognitive attitudes and affective attitudes, and the respondent's age, sex, level of education, type and home computer availability. The results are presented in Table 1.

It can be noted from the table that there were low positive correlations between computer knowledge and computer attitudes. Similar correlations were found between age and computer knowledge and attitudes. Level of education showed low, positive relations with knowledge and attitudes. Being a student or teacher related more strongly with computer knowledge and attitudes than did being an administrator, media specialist or other educator. There was no relation between having a computer in the home and computer knowledge or attitudes.

Discussion

From this portion of the study, it can be seen that students and educators tend to be functionally literate about computers and have generally positive cognitive and affective computer attitudes. Computer knowledge seems to be positively related to attitudes, age, level of education and being a student or teacher. Similar variables seem to be related to computer attitudes.

Note

In the program presentation, other information pertaining to the use of computers will be examined. These data will include the types of computers used, types of uses for computers, and the interrelations of these variables. Included also will be a discussion on the willingness of students, teachers and others to copy software.

TABLE 1

SIMPLE CORRELATIONS BETWEEN INDIVIDUAL BACKGROUND VARIABLES AND COMPUTER KNOWLEDGE AND ATTITUDES

VARIABLES	COMPUTER KNOWLEDGE	COGNITIVE ATTITUDES	AFFECTIVE ATTITUDES
Computer Knowledge	-----	0.316	0.359
Cognitive Attitudes	0.357	-----	-----
Affective Attitudes	0.316	-----	-----
Age	0.289	0.350	0.326
Sex	0.159	0.017	0.021
Levels of Education			
High School	0.215	0.265	0.243
University	0.158	0.185	0.155
Type Respondent			
Student	0.222	0.285	0.267
Teacher	0.164	0.233	0.201
Administrator	0.032	0.063	0.093
Media Specialist	0.082	0.080	0.094
Other Educator	0.087	0.075	0.061
Computer in Home	0.011	0.065	0.083

Documentation - A Key To Success

Carolyn Meinhardt
Wordware
Rt 1 Box 74
Howard Lake, MN 55349

Documentation serves three major purposes; TRAINING - It is a permanent teacher always available for training personnel, REFERENCE - It is the ready source of help and reminders, STANDARD - It provides the common denominator of understanding of how the system is supposed to work. This paper will describes how good documentation is written, what kinds of documentation should be written and why, who should write it and when. Included will be discussions on content, techniques and writing style.

Documentation is a broad term which describes any written instructions that preserve information. Today's commercial computer software developers and their users find themselves reading and writing many kinds of documentation. Although the buyers of commercial software place a high value on the quality of the product's documentation, many non-commercial software developers place little importance on writing documentation for themselves or their users and many organizations do not document their own use of computer software.

PURPOSES FOR DOCUMENTATION

Written instructions or documentation serve three primary purposes.

Training - Documentation is a kind of permanent teacher. If the material is complete and logically constructed, training may be accomplished solely by using the written material, or it may be supplemented with a teacher. Remember that any information stored only in the head of a teacher and not found somewhere in written form, will be unavailable when the teacher leaves.

Reference - Documentation is a readily available source of help. It can provide a refresher course on how to do an infrequently needed task, or assist in determining how the product can be put to a new use.

Standards - Often the definition of software "not working" is that it isn't doing what it is supposed or expected to do. Documentation provides a written description of what can be expected from a fully functioning product. It may reveal that the product was never intended to function as the user desires, or it may prove that something is amiss and needs to be corrected. In this way, documentation can provide a common denomiator of understanding between the user of the product and the creator or maintainer of it.

THE VARIOUS KINDS OF DOCUMENTATION AND WHY THEY ARE NEEDED

There are many stages in the development and implementation of a software product or system, and documentation should accompany each stage. Some of it will only be used by the developers or maintainers, and some will be intended primarily for the end user. Organizations such as school districts will find that internal procedural documentation, something that cannot be provided by the manufacturer, will need to be produced by school staff.

System Proposals - Developers of software create products which are intended to meet needs of anticipated users. Those needs are described in detail in a document called a Statement of Needs. The Proposal suggests a solution to those needs by describing the features of the proposed product, including the required input information and procedures, the processing that will occur, outcomes or reports that will be available, realistic operating and turnaround times, and all features. Changes can be expected in every project, but the System Proposal should remain the official description from which the user can learn what the product or system will do.

Systems Design - In a project, users are the experts on what the system should do, but computer science professionals or analysts determine the plan for the most efficient way of getting the computer to do it. Most analysts dislike producing detailed documentation but project managers who realize the impermanence of employees, also realize that it is necessary that this kind of documenting be done. Besides insuring that a project will not be disasterously set back because of resignations or illness, the continual reviewal of the Systems Design against the work in progress will insure that everyone is working towards the intended outcome.

Technical Descriptions - This documentation includes information about how the computer will process the software, what the data records will contain, where they will be located, and other information which will be vital to future software maintainers. It is the story of how the programs were written. It may or may not be made available to the purchasers of prepackaged software, but it will be vital to anyone who tries to modify the software in the future.

Operating Manual - An operating manual tells how to operate the computer in order to obtain the desired results from the software. Mainframe computers typically have people whose job is to run the machine and the software. However, when microcomputers are used, the users are often also the operators. In this case the operating instructions may be included in the User Manual.

User Manual - This documentation will initially teach how to use the product, and then later provide reference information. It may include input forms, step-by-step guides, tutorials, explanations of the purposes of features, reference and summary appendices, etc. For customized products, there may also be an explanation of the purpose of the product, i.e. what it will be used for in the organization, and related policies. The documentation for microcomputer software will stress "what keys to press" as well as how to make use of the system.

Procedures Manual - When a product is used to facilitate a function of an organization, such as recording attendance for a school, then it becomes extremely important that a common set of standards and procedures be used and described in writing. Information and procedures which should be included are schedules (daily, weekly, etc.), input forms and procedures, verification procedures, regularly produced reports including the parameters and file names used to produce them, distribution procedures, equipment hook-up, maintenance and back-up procedures, and anything else which an unoriented person with only a limited amount of background information would need in order to step in and get the work out.

DOCUMENTATION AUTHORS

It is possible for documentation to be written by someone who has very little direct connection with the development or use of the product. The most important quality of the person writing the final documentation is having good communication skills. The ideal writer would be someone who has deep knowledge of the product, and is also a good communicator. The next best choice may be for the knowledgable person to write a draft which is smoothed out, filled in and simplified by the official writer.

When the end user is involved in a development project, it is best to have the Statement of Need written in the language of the user. The document may be a combined project of development staff and users, but it should be written for a non-technical audience. Terms should be defined so that everyone has a clear understanding of any jargon.

The Systems Design and Technical Descriptions are written to be used by technical staff, and should be concise, well organized, and conform to the development standards of the organization. Analysts and programmers will most often be the authors, although many departments have found that a documentation staff person can produce better documentation, more quickly, leaving the programmers free to program. This same group may write the Operating Manual for mainframe computer products.

User Documentation and Operating Manuals for microcomputer software products, should be written by someone who knows how the product can benefit the user as well as how to "make it work." Users often write the best user manuals, although they should be guided by standards. Procedure Manuals must be written by a user, preferably someone who has a broad organizational viewpoint and can relate how each operation interfaces with the others.

THE USER MANUAL

Most software product evaluations include comments about the usefulness of the product's user documentation. Although the trend with much pre-packaged microcomputer software is do provide self-help screen prompts, these hints are most helpful for users who already know what they want to do with the product. Most users of either micro or mainframe computer products will find that the best documentation not only tells them how to accomplish a task, but also tells why they are doing it, what features can be used to accomplish it, and suggests the fastest way of completing the task.

Determining the Audience - Documentation must always be written with the audience in mind. What is their level of technical knowledge? For microcomputers, have they ever used the machine before, or will general operating instructions also need to be included? What information and help are they looking for? Should the manual include overviews for management, details for first-time users, examples suggesting real-life situations, technical references, or all of these and perhaps more? What language does the audience speak? Technical jargon is unnecessary and often undesirable in good documentation. Technical terms which are required for communication, should be defined in each document, on the assumption that if they are not defined, someone will be confused.

Organizing the Contents - The organization of the manual will largely determine its usefulness. Instructions can be organized by system feature or function (ex: The Master Schedule Form or The Page Offset Command) or by task (ex: "How to Define the Master Schedule" or "Using Letterhead Stationary".) Generally, information which is explained by task groups several features together in a way which makes more sense to the uninitiated. Task oriented manuals especially need excellent topic indexes and summary appendices. Each section should include an introduction - what the feature or task will accomplish, a body which includes clear instructions, and a brief summary which can serve as a refresher course later on. Manuals which describe systems such as school scheduling packages, can benefit from a separate step-by-step guide which refers the user to the specific section in the manual dealing with that topic. Samples of screens, forms and reports are extremely important and must be large enough to be legible. Samples take up a lot of room but are very effective communicators.

Technical Writing Considerations - Documentation writing and production is the place to use word-processing. The efficiences are most realized when the writers are able to write on the same system that will be used for archiving and production. Microcomputers make this easy since the floppy disks can be easily updated, shared and stored, and eliminate the need for re-keying the text. The many print features available in the more sophisticated products will make the manuals attractive and easy to read. All manual pages should be page-numbered and have the section title at the top. Many organizations also include a logo and manual title on each page. Leave plenty of room in the margins for users to make notes. If the manual will seldom be revised, or will be reprinted in its entirety when revised, plan to print on both sides of each sheet of paper. If the manual is subject to frequent updates, use only one side, and print the revision date on each page. The manual will be thicker but easier to update. When the time comes for revision, the replacement page should reflect the new revision date, and some kind of identification in the margin as to what was changed.

THE PARTS OF A COMPLETE MANUAL

Not every manual will include all of the parts listed here, but each item adds value to the documentation.

Title Page - Include the last printing date of the entire manual.

Preface - This is a brief description of the product and its purpose.

Table of Contents - Include every topic heading, preferably in outine form.

Revision Checklist - This is a list which includes every page in the manual with its most recent revision date. Pages with the same date can be listed as a range of page numbers.

Introduction - Explain what is in the manual and how to use it.

Information Sections - These may include machine operating topics, a step-by-step guide, topics arranged by function or task, etc.

Appendices - Include command summaries, quick reference lists, cross-reference lists, samples of forms, etc.

Glossary of Terms

Index of Topics

New writers may find help in planning the layout of a manual from the many books available. It is important that some kind of standard be used, so that the manual(s) will be easy to maintain. Most large commercial software firms or companies with their own data processing centers have sets of standards which they are willing to share.

QUALITY ASSURANCE TESTING AND DOCUMENTATION

The writing of the user documentation and final quality assurance testing can go hand-in-hand. Some development groups do system testing while the user documentation is being written, and then use the documentation as the standard for a final independent test of the system conducted by non-development staff. This method often results in a useful critique of the documentation as well as uncovering bugs which otherwise first show up when the users get the product.

THE COST OF PRODUCING DOCUMENTATION

Good documentation is not cheap. Besides the labor involved in doing the writing, time must also be allocated for reviewal and rewriting as needed. Where word-processing is not used, additional time will be needed for typing, retyping changes, and proofing. Printing costs will vary depending on the length of the manuals, distribution and frequency of use. Documentation of systems which undergo regular changes can become quickly outdated - procedures and resources must be established for producing revision updates. It will be important to maintain accurate lists of who has which manuals.

THE COST OF NOT HAVING DOCUMENTATION

The cost of writing and having good documentation must be considered against the cost of not having it. Without documentation, training always requires the time of two people - the teacher and the student. The trainers must themselves be trained as changes to the system are developed. Users must call a person if they have questions and the person may not always be available. Many mistakes occur because of guessing. There is no standard for performance of the system, so trouble-shooters of reported problems must determine if the user made a mistake, or if the system has a bug, or even if the system is supposed to have the capability in question. Information on how to use a product can disappear along with former employees. The list of problems is endless, and the real cost in labor can quickly exceed the "savings" from not taking the time to write the documentation.

Computers are becoming very user friendly, but as their use grows, so does the need for training, reference and understanding. Commercial softwares' quality is often judged on the useability of the documentation which comes with the packages. Customized software developed for the use of schools should be given the same kind of consideration. The time and money spent will always justifiable because as long as there are questions about how to use systems, there will be a need for good documentation.

Educational Computing In Britain

John W. Brine and E. Marcia Johnson
The Ontario Institute for Studies in Education
Measurement, Evaluation and Computer Applications
Toronto, Ontario, Canada

Educational computing is a rapidly expanding field, and while it is difficult for educators to keep abreast of developments in their own countries, it is even more difficult to learn of parallel work in different countries. In September 1984, the authors were sponsored by the British Council to visit several universities each to investigate the state of educational computing in Britain. This paper provides a brief overview of the Microelectronics Education Programme (MEP) in England and the Scottish Microelectronics Development Programme (SMDP).

Background

The Microelectronics Education Programme (MEP) was introduced by the Conservative government in 1980 with the stated aim of helping schools to "prepare children for life in a society in which devices and systems based on microelectronics are commonplace and pervasive" (Fothergill, 1984, p.1). This programme, intended for schools in England, Wales, and Northern Ireland, was to receive financial support until 1984; however, funding has since been extended until 1986 To date funding for the programme has been approximately L20,000,000. The headquarters for the MEP is in Newcastle-upon-Tyne in North-Eastern England.

Scotland has its own programme, called the Scottish Microelectronics Development Programme (SMDP) with headquarters in Glasgow, Scotland. This programme, also established in 1980, had a four-year mandate and a budget of L1,000,000.

The Microelectronics Education Programme

Some of the underlying assumptions of the MEP are that:

1. schools are expected to respond to changes in society by amending the content and approach of existing individual subjects in the curriculum and by developing new topics to teach about information technology;
2. methods of teaching and learning should make use of microcomputers in such ways that new dimensions will be added to the relationships both between teacher and class or between teacher and pupil;
3. microcomputers should be used to develop an individual pupil's capacity for independent learning and information retrieval;
4. microcomputers should be used to aid the physically handicapped.

Scope of the Programme

The initial focus of the programme was on 16-19 year old students in non-vocational courses. Specifically, research was to be encouraged to investigate the most appropriate ways of using the computer as an aid to teaching and learning, as a guide to the individual child, as a learning aid for small groups of children, or as a system which involves the whole class. The stated curriculum foci were mathematics, the sciences, craft/design technology, geography, and business with some attention being given to careers education, languages, and the humanities. Special education, for those handicapped in some way, has received significant attention.

New topics to be developed for the curriculum included:

- microelectronics in control technology;
- electronics and its applications in particular systems;
- computer studies;
- computer linked studies, including computer aided design;
- word processing and other 'electronic office' techniques;
- use of computer as a means of information retrieval from databases

Programme Activities

The MEP concentrated research funding in three major areas: curriculum development, teacher training, and resource organisation and support.

Curriculum Development

In the area of curriculum, MEP funding was intended to facilitate the development of materials making use of computers in the teaching of traditional subjects, and supporting the teaching of the new curricula defined above. It was also expected that a national software industry and dissemination medium would be developed. Moreover, it was recognised that in the longer term, the issue of how to evaluate new curriculum specifically dealing with information technology needed to be addressed.

One major purpose of the MEP has been to ensure that curriculum research efforts are supported in many different locations in Britain and are not duplicated from one area to another. Consequently, it was intended that the most substantial allocation of resources for curriculum development would be made to research groups collaborating with two or more local education authorities (LEA). Perhaps most importantly, research proposals from classroom teachers have been encouraged as it was assumed that practising teachers should contribute to all curriculum development projects.

In addition, in order to advance a general understanding of microelectronics applications, the MEP is commissioning a number of supporting documents, including case studies and examples of good practice throughout the curriculum. It was felt that such research would stimulate informed comment which could thus be reflected in examination syllabuses for new topics in the curriculum.

Teacher Training

It was understood that in order to apply information technology effectively in the classroom, teachers required not only information about microelectronics, but also new professional skills. If an innovation is to gain acceptance in the educational field, teachers must be adequately trained and then kept informed of further developments. To this end, resources needed to be devoted to train teachers, but since the MEP does not envision itself funding such training indefinitely, the programme has offered 'pump-priming' support for the establishment of a regional network of in-service training centres. The object of establishing these regional centres has been to enable groups of authorities to coordinate their resources for in-service training. The training of pre-service and in-service teachers was planned initially in three ways:

1. Short courses (1-3 day's duration) to develop general awareness and familiarisation with different types of equipment;
2. Short specialist familiarisation courses (1-2 week's duration) for enthusiastic teachers wishing to modify their subject teaching to include new topics, for example, commerce teachers learning word processing;
3. Longer specialist courses (up to three months duration) aimed at teachers requiring additional training in particular fields, for example, craft/design teachers wishing to learn more about electronics.

While these courses were intended to ameliorate an immediate need, the longer-term issues of designing diploma and degree courses in microelectronics were also to be considered.

Regional Resource Organisation and Support

The development of new curriculum materials would be relatively futile if no mechanism existed to disseminate such information. Rather than attempt to establish new communication networks, the MEP decided to build on the foundations of the LEA advisory services by establishing a set of pilot information centres to serve groups of LEA's. There have been twelve centres established in England, and one each in Wales and Northern Ireland. Their roles are:

- to exchange information about curriculum materials and computer software with others regional centres in the network;
- to exchange information with in-service training centres;
- to form links with LEA advisory services;
- to provide some facilities for in-service training;
- to keep in touch with curriculum development projects;
- to disseminate software;
- to provide access to 'debugging' facilities;
- to organise local group sessions and support for distance education;
- to form links with manufacturers;
- and to provide information to the community, such as:
    1. materials and equipment available and their evaluations;
    2. places where materials and equipment could be viewed and tried;
    3. known developments in teaching materials;
    4. a small collection of reference texts;
    5. and a software library.

In order to accomplish these goals, MEP resource allocation priorities included the establishment of a network of pilot regional information centres, the collection of information on microelectronics activities in the regional area, the investigation of different methods of software distribution, and work on the formulation of standards.

Provision of Microcomputers for Schools

While the aim of the MEP is to promote a wider understanding of the potential role of microelectronics in education, the acquisition of equipment was viewed as being outside of the programme's scope. However, the Department of Industry made funds available to schools with the aim that there would be at least one microcomputer in every secondary school by end of 1982 (accomplished). Schools already possessing microcomputers were not given priority for this funding.

This aim having been satisfied, the programme has been expanded to include primary schools.

The Scottish Microelectronics Development Programme

The SMDP has been a much smaller project (both in funding and in scope) than its counterpart, the MEP, in the rest of Britain. Its major goal has been the generation of good software and the dissemination to Scottish educators of both software and information about appropriate educational applications. As part of the programme, microcomputers were loaned to 68 schools and other educational institutions. In 1982, a library of SMDP software was established and software is freely distributed to Scottish schools.

In 1983 a national coordinating body, the Microelectronics in Education Committee (MEC) was established with representation from local education authorities, teacher training and curriculum development bodies, teacher unions, and institutes of higher learning, such as universities and further and higher education colleges. Their mandate has been to develop a national software library, a national information service, and a research and development function.

Supported Software Library

Software is developed through a process of formative evaluation and feedback. SMDP programmers use specifications written by classroom teachers who then test the product in their classrooms. The programs are then amended by the programmers. Documentation is written to SMDP-developed standards and includes notes for students, teachers, and technicians.

Three regional distribution centres maintain complete sets of the Supported Software Library programs and receive new and updated software regularly. The programs are intended for the Apple, BBC, Commodore Pet, Cromemco, RML (Research Machine), and Sinclair Spectrum microcomputers.

Information Dissemination

At the headquarters in Glasgow, the SMDP maintains a display area containing relevant books and periodicals; however, the area can only be used if a appointment to browse has been made. The SMDP also publishes a periodical for distribution throughout Scotland. *Phase Two* includes articles about successful implementations and reviews of software, books, and hardware.

As in the MEP programme, there are regional centres in Scotland that maintain frequent contact with the SMDP and exchange information with each other on a regular basis. These centres have also been working to establish international contacts.

Scottish Community Education Microelectronics Project (SCEMP)

An associated project is the SCEMP which aims to increase general awareness of the potential scope of the microcomputer as a community resource. This project is exploring how microcomputers can be used:

- to provide information within a variety of community settings;
- and, to create individualised learning opportunities within the community (About the SMDP, 1984, p. 7).

Other SCEMP activities include the creation of a database describing appropriate software and the development and evaluation of appropriate computer programs for use in community education.

Associated Software Projects

The SMDP is coordinating the Primary Software Development Project to produce software appropriate for primary schools. As with the MEP, the software development and creation process is distinguished by its use of a formative evaluation and feedback cycle. In May 1983 teachers were identified to participate in the project and met on a regular basis:

- to discuss and prepare program specifications;
- to discuss programs written by the SMDP and their field tests;
- and, to write and discuss program documentation

Software produced by the SMDP will be maintained and distributed through the Supported Software Library.

Another software production project is housed in Edinburgh; its mandate is also to produce software for primary schools.

Success of the Programmes

The MEP has been very successful in introducing computers into the school system. Although it was begun several months earlier than the MEP, the SMDP has had less success in distributing computer technology. This is probably due to the inherent centralisation in the Scottish implementation plan. By September 1984, this problem had already been recognised and attempts were being made to increase local involvement to the higher level found throughout the rest of Britain.

We found a great deal of evidence of both collaborative and action research involving university faculty, researchers, and teachers. Moreover, teachers are frequently involved not only in the planning and carrying out of classroom-based research, but also in the interpretation of project results.

Educational researchers in Britain have always emphasised qualitative methodologies more than their counterparts have in North America. In researching the use of computers in education, ethnography, case studies, and formative evaluations have been well-suited to collaborative studies with teachers and in designing software. However, several researchers still perceive a need for more detailed, well-constructed, and carefully reported case studies in the area of computer applications in education.

Software development in England is concentrated in three major centres: Five Ways Software in Birmingham; Chelsea College in London; and the College of St Mark and St Matthew in Plymouth. Software is seldom of the CAI drill-and-practice variety, but is intended to be more open-ended and functional at several grade levels and across subject areas. Packages are marketed through publishers who must ensure adequate documentation. Moreover, most packages are developed to run on more than one computer, including the RML, the BBC, and Apple computers. While all of the software development centres mentioned above employ a young staff (for example, a number of Five Ways employees were recent 'A' level graduates) and develop and test software in collaboration with subject area teacher specialists, the software centres find the UK market relatively small. Five Ways software is already being marketed in Europe (in translation) and is about to be sold in North America.

Conclusion

Although this paper can only describe briefly educational computing in Britain, there are a couple of points that can be made in conclusion:

1. The MEP approach to educational computing is extremely well-organised and concentrates on establishing new, and enhancing existing, human links.
2. Research into computer applications in education is not the exclusive domaine of universities, but rather it includes a variety of people from the wider educational system.
3. The major focus of the programme is not on the procurement of increasingly sophisticated hardware, but rather on the development of appropriate pedagogical techniques to utilize existing resources. The emphasis is on applications and education, not on technology in isolation.

REFERENCES

About the Scottish microelectronics development programme. (1984). SMDP, Glasgow: Scotland.

CET information sheet: Microcomputers in secondary education. (1984). Council for Educational Technology, Newcastle-upon-Tyne: England.

Fothergill, R. (1981). Microelectronics education programme: The strategy. Department of Education and Science, Stanmore: England.

MEP information sheet: Microcomputers in primary education. (1982). MEP, Newcastle-upon-Tyne: England.

Anderson, J.S.A. et.al. (1983). The start of MEP INSET 1981-1983. MEP, Newcastle-upon-Tyne: England.

Characteristics of the Periodical Literature in Computer Applications to Education

Irvin Rubincam and Gail Somer
The Ontario Institute for Studies in Education
252 Bloor Street West, Toronto, CANADA M5S 1V6

A search of the computer applications (CA) to education literature for 1982 identified 503 articles. These articles were analyzed according to the following characteristics: occupation and place of residence of the authors; when (how far back in time) citations are being made; and the most frequently cited journals, authors and publications. Some observations are offered concerning missing citations and the current state of the CA literature.

Introduction

The old adage about it being difficult to see the forest for the trees comes to mind when one tries to understand the complex and rapidly growing field of computer applications to education. Instruction in virtually all areas of knowledge can be assisted in one way or another by the use of the computer. Educators, subject-matter experts, psychologists, and computer scientists all contribute to the field from their own perspectives. Given such diversity, it is not easy to determine what is the essential and core knowledge of the 'discipline' of computer applications. Yet a body of theoretical and applied knowledge must be developed and widely understood if there is to be substantial progress in fulfilling the potential instructional value of computers. The alternative is to perpetuate the fragmentation of knowledge among the various subject areas and disciplines, making it more difficult to share a common theoretical perspective.

This report is a step in the direction of understanding the larger picture. The objectives are as follows:

1. To develop a methodology for examining publications in computer applications to education.
2. To assess the current state of the 'discipline' of computer applications with regard to information sharing and theory building.
3. To establish bench marks for subsequent evaluations of the computer applications literature.

Characteristics of the Data Base

A search was conducted of the English language periodical literature in computer applications for the year 1982. From 71 periodical sources 503 articles were identified. The following journals contributed the greatest number of articles: *The Computing Teacher* (46 articles); *Educational Technology* (36 articles); *American Annals of the Deaf* (30 articles); *Computer Education* (34 articles); *Classroom Computer News* (24 articles); *Computers and Education* (21 articles); *AEDS Journal* (18 articles); *Compute!* (16 articles); *Journal of Educational Technology Systems* (16 articles); *NASSP Bulletin* (16 articles); *Creative Computing* (13 articles); and the *Journal of Computer-Based Instruction* (13 articles).

Authors of the Articles

In 83% of the articles, the occupation and institutional affiliation of the author(s) were identified. The percentages of authorship in major categories were as follows:

55% Teachers, researchers, students at the post-secondary level
15% Teachers and administrators at the elementary and secondary level
10% Educators at the regional, state, provincial, or national levels

10% Persons from business or publishing
10% Others (medical, military, private individuals)

That the majority of articles (55%) are authored by teachers, researchers and students at the college and university levels is not particularly surprising. Post-secondary institutions have had a longer exposure to CA than the public schools. Most research is carried out in college and university settings. Faculty members must publish to receive tenure and promotion. Educators at the elementary, secondary, and governmental levels, despite having heavy workloads with limited time for reflection and writing, contributed 25% of the articles.

Where the Authors Live

In 427 of the articles the place of employment (or residence) of the author could be identified. U.S. authors produced 329 articles (77% of the total). All but 4 states contributed at least 1 article. One state, Massachusetts, produced significantly more articles than would be expected on the basis of its population. Authors in 21 other countries contributed a total of 98 articles, including 41 from the U.K. and 18 from Canada.

Citation Analysis

Of the 503 articles, only 240 (48%) contain a list of references. This is significant in itself, and reflects the dichotomy in the CA literature between articles in scholarly and refereed journals (which generally include reference lists) and articles in the popular magazines (which generally do not). Still the total number of references (2691) is of sufficient size upon which to base some general conclusions.

The following questions were asked of this list of references:

1. When (how far back in time) are citations being made?
2. What are the most frequently cited journals?
3. Who are the most frequently cited authors?
4. What are the most frequently cited articles?

Chronological Distribution of Citations

The Social Sciences Citation Index (SSCI) publishes annual figures on the cumulative percentage of citations in the social sciences literature for each year from the present to 1950. This information has been graphed in Figure 1. It can be seen that, by 1950, 92% of the citations have been accounted for. CA writers on average cite more recent publications, so that by 1950, 99.4% of CA citations have been accounted for. It remains to be seen whether the CA curve will move closer to the SSCI curve over time.

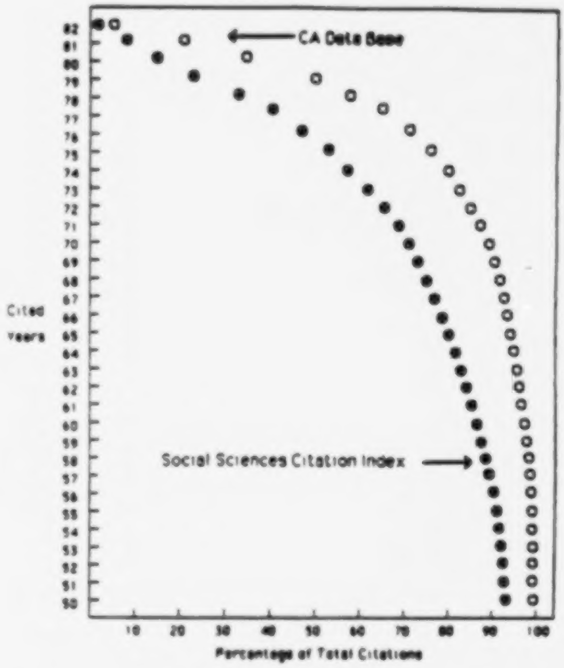


Figure 1 Chronological Distribution of Citations
SSCI and CA Data Bases

BEST COPY AVAILABLE

Journal Citations

The nine most frequently cited periodical sources are the following:

Educational Technology (103 citations)
ERIC (73 citations)
AEDS Journal (35 citations)
Creative Computing (30 citations)
American Annals of the Deaf (28 citations)
Journal of Educational Psychology (26 citations)
Journal of Computer-Based Instruction (23 citations)
Computer (21 citations)
Phi Delta Kappan (18 citations)

Most Frequently Cited Authors

The 2691 references were used to tabulate the most frequently cited authors in the CA literature. The following authors received 10 or more citations:

Seymour Papert (26 citations)
Robert M. Gagne (18 citations)
Andrew Molnar (13 citations)
Richard C. Atkinson (12 citations)
Benjamin Bloom (11 citations)
Patrick Suppes (11 citations)
Thomas W. Malone (10 citations)

Most Frequently Cited Books and Articles

Books or articles cited three or more times are listed below:

Papert, Seymour. *Mindstorms*. New York: Basic Books, 1980 (12 citations).

Molnar, Andrew R. The next great crisis in American education: Computer literacy. Washington, D.C.: National Science Foundation, 1978. [ED 191 733] (1 citation)
also *Journal of Educational Technology Systems*. 1978-79, 7, 275-283. (1 citation)
also *AEDS Journal*, 1978, *12*, 11-20. (1 citation)
also *THE Journal*, July/August, 1978. (4 citations)

Malone, T.W. What makes things fun to learn? A study of intrinsically motivating computer games. Palo Alto, California: Xerox Palo Alto Research Center, 1980. (6 citations)

Baker, Frank B. *Computer Managed Instruction: Theory and Practice*. Englewood Cliffs, N.J.: Educational Technology Publications, 1978. (5 citations)

Gagne, Robert M., & Briggs, Leslie. *Principles of Instructional Design* (Second Edition). New York: Holt, Rinehart & Winston, 1979. (5 citations)

Evans, Christopher. *The Micro Millenium* (alternative title: *The Mightly Micro*). New York: Washington Square Press, 1979. (4 citations)

Gagne, Robert M. *The Conditions of Learning* (Third Edition). New York: Holt, Rinehart & Winston, 1977. (4 citations)

Newell, Allen, & Simon, Herbert A. *Human Problem Solving*. Englewood Cliffs, N.J.: Prentice Hall, 1972. (4 citations)

Briggs, Leslie, & Wager, W.W. *Handbook of Procedures for Design and Instruction* (Second Edition). Englewood Cliffs, N.J.: Educational Technology Publications, 1981. (3 citations)

Moursand, David. *Introducation to Computers in Education for Elementary and Middle School Teachers*. Eugene, Oregon: International Council for Computers in Education, 1981. (3 citations)

Notable Missing Citations

Missing from the list of frequently cited authors are a number of important persons in education and related fields. Among the more significant omissions are the following:

B.F. Skinner is one of the most influential psychologists of the 20th century. In recent years, the attention of educators has shifted toward developmental and cognitive psychology. It is perhaps not surprising, then, that Skinner is not included on the most cited list. An examination of the Social Sciences Citation Index reveals, however, that interest in Skinner's 1968 book *The Technology of Teaching* remains at a reasonably high level (18 citations in 1982).

Jean Piaget is another important psychologist who is surprisingly absent from the citation list. In 1982, SSCI listed over 1000 citations of Piaget's writings. Piaget's theories about the stages in a child's acquisition of knowledge and understanding would seem to be relevant to many discussions of CA, particularly at the elementary and intermediate levels. Seymour Papert, a former student of Piaget, acknowledges his debt to Piaget's ideas in *Mindstorms*. Only three citations of Piaget's work were found among CA writers.

John Dewey is a third major writer to receive little attention in the CA literature. One of his most influential ideas concerns the importance of active problem solving and 'learning by doing'. This idea forms an important part of the rationale for CA in general, and, in particular, computer simulations of processes in the real world. While Dewey is an historic figure, whose major works were first published at the beginning of this century, his writings still receive a substantial number of citations (227 in SSCI in 1982).

Critics of the Role of Computers in Education

Over the years there have been few books critical of the use of computers in education. One notable exception is *Run Computer Run* written by Anthony Oettinger and published in 1969. It is his view that, although instructional technology can contribute to genuine improvement in education, it is currently (1969) being "force fed, oversold, and prematurely applied." Perhaps such ideas are no longer relevant. In any case, interest in this book has declined over time. Only two citations were found in the CA data base.

A second book of a critical nature which might be of interest to CA writers is *Computer Power and Human Reason*. The author, MIT computer scientist Joseph Weizenbaum, questions the value (and morality) or work in artificial intelligence techniques. CA writers might want to consider Weizenbaum's concerns. While interest in the book remains high (22 citations in SSCI in 1982) there was only 1 citation of Weizenbaum in the CA data base.

Observations on the CA Literature

The periodical literature in CA (as of 1982) emphasizes practical matters concerned with the implementation of computers in education. Many writers focus on their own local situation. Even discussions of issues of a more general nature (computer literacy, the changing role of the teacher, etc.) are based more upon personal opinions rather than a consideration of a body of theoretical writing.

There are a few exceptions to these general trends. For example, faculty members at Florida State University (Robert Gagne, Leslie Briggs, Robert Reiser and others) have made significant contributions to the fields of educational psychology and instructional design theory. Their publications contain extensive bibliographies, (including many references to previous and current work by Florida State faculty and graduates (e.g., Hannum & Briggs, 1982). This 'community of scholars' is found in many fields but is, as yet, rare in CA.

Another sign of a mature discipline, which is largely lacking in CA writings, is the effort by a researcher over a significant period of time to solve a particular problem or class of problems. This phenomenon can be most clearly seen in the self-citations of the researcher. Study A is published, then study B which cites study A, then study C which cites studies A and B, etc. Each study contributes new knowledge but reveals unanswered questions and the need for further studies. There is at least one clear case of this type of sustained research over time in the CA literature. In 1975 Robert Tennyson and his colleagues at the University of Minnesota began the development of the Minnesota Adaptive Instruction System (MAIS). This system has served as both a delivery system for instruction and a means of controlling various experimental conditions. The experiments are designed to test four variable features of an 'intelligent' computer-based instruction system. These include control over: the amount of instruction, the sequence of instruction, instructional display time, and advisement information.

Tennyson, Christensen, and Park (1984) describe the results of numerous experiments conducted since 1977 using the MAIS. The variable of advisement information will serve as an example. Advisement means giving students information on their progress and advice on appropriate learner-control decisions. Four studies have examined the effect of advisement on student performance. Tennyson & Buttrey (1980) found that providing students with diagnosis and prescription information improved learning when compared with program-control and learner-control (without advisement) strategies. Tennyson (1980) found that advisement with learner-control is more effective than adaptive program control. Tennyson (1981) reported that advising students in a learner-control condition improves their concept- and rule-learning performance. Finally, Johansen & Tennyson (1984) showed that a student's perception of the learning task is enhanced by advisement during the learning of rules. In summary, Tennyson's systematic and long-term approach to research in CA can serve as a model for other researchers in this field.

The 1983 CA literature is currently being analyzed. This information will be compared with the 1982 data reported in this article in an effort to understand how the 'discipline' of computer applications is progressing over time.

References

Hannum, W.H., & Briggs, L.J. How does instructional systems design differ from traditional instruction? *Educational Technology*, January, 1982, 9-14.

Johansen, K.L., & Tennyson, R.D. Effect of adaptive advisement on perception in learner controlled, computer-based instruction using a rule-learning task. *Educational Communication and Technology Journal*, 1984, *3*(1), 226-236.

Oettinger, Anthony. *Run, Computer, Run*. Cambridge, M.A.: Harvard University Press, 1969.

Skinner, B.F. *The Technology of Teaching*. New York: Appleton-Century-Crofts, 1968.

Tennyson, R.D. Instructional control strategies and content strategies as design variables in concept acquisition using computer-based instruction. *Journal of Educational Psychology*, 1980, *72*, 525-532.

Tennyson, R.D. Use of adaptive information for advisement in learning concepts and rules using computer-assisted instruction. *American Educational Research Journal*, 1981, *4*, 425-428.

Tennyson, R.D., & Buttrey, T. Advisement and management strategies as design variables in computer-assisted instruction. *Educational Communications and Technology Journal*, 1980, *28*, 169-176.

Tennyson, R.D., Christensen, D.L, & Park, S.I. The Minnesota Adaptive Instructional System: An intelligence CBI system. *Journal of Computer-Based Instruction*, 1984, *11*(1), 2-13.

Weizenbaum, Joseph. *Computer Power and Human Reason*. San Francisco: W.H. Freeman, 1976.

Sexism, Computers and Young Children

Gary Woodill
Ryerson Polytechnical Institute
Toronto, Ontario

Karen Anderson
King's College
London, Ontario

Judy Bernhard
Ryerson Polytechnical Institute
Toronto, Ontario

Recent Studies have indicated that computers in the classroom are often dominated by the boys. This article examines the literature on this issue, with particular reference to preschool children. The preliminary results of a pilot study using a computer in a day care centre are described, and possible explanations for gender differences in computer use are given, along with some strategies for dealing with this problem.

A number of recent studies (Kiesler, Sproul and Eccles, 1983; Fisher, 1984; Lockheed and Frakt, 1984; Miura and Hess, 1984; Rampy, 1984) have reported that in the higher grades of school, boys use the computers in the classroom much more than the girls. When the focus turns to younger children, the picture is not as clear. While Kiesler et al. (1983) describe a Michigan preschool where "the boys created a computer club and refused to let the girls have access to the equipment", and Silvern and collegues (1982) found that 85% of the computer users in a group of children ages 4 to 8 were boys, others have not found a significant sex bias in computer use in the early years. For example, Piestrup (1981) observed more girls than boys using the computer in a preschool classroom, except on the day when the visitor was a male programmer. In spite of a large literature which indicates that boys have superior visual-spatial abilities, a recent study of three and four year old children (Sprigle and Schaefer, 1984) showed no gender effect on programming graphics using LOGO.

In May 1984 a computer system (Commodore 64) was introduced to the Early Learning Centre at Ryerson Polytechnical Institute in the classroom with the oldest children, ages 4 to 5. It was received enthusiatically by the children, and with some mixed feelings by the staff. Because of the positive reactions of the children, it was decided to leave the computer in the centre, as one of several learning areas in which the children could choose to work for a one hour period in the morning.

During September 1984 the supervising teacher recorded the children's choices beside their names, giving a profile of which children chose the computer over other areas of the room. During a three week period, the computer was chosen by a girl only once, while the boys chose 16 times. With 3 girls and six boys in the class, this results in a pro-rated ratio of 1 time for girls to 8 times for boys.

In late November the use of the computer was expanded to include the younger children in the centre. It was available for an hour in the late afternoon when all 36 children in the centre were in one large room. The children who chose to work on the computer had their names placed on a waiting list, and worked two at a time at the computer. The children could ask for second turns when everyone was finished their first turn.

Because the 4 to 5 year olds who used the computer in the mornings, could also ask to use it in the afternoon period, we were able to compare their results in September under a new set of conditions including a change of teacher, a different room and the presence of younger children. In addition, we were able to observe frequencies of requests for the computer for three different age groupings over a 10 day period.

The number of boys and the number of girls was each divided into their respective frequencies of requests for turns at the computer (in order to equalize for differing numbers of boys and girls), producing an average number of requests for each sex. The higher average was then divided by the lower average to give a ratio of girl-boy use which could be compared across groups.

The results showed no major gender differences for any age group in the number of times the children asked to use the computer. The older 4 to 5 year olds had a very close girl-boy ratio (1 to 1.1). The younger 4 year old group had an identical ratio. In the the 2 to 3 year old group there was the same ratio except that the girls had the edge.

While more study with larger numbers of children is obviously needed, the results of this pilot study raise a number of points for discussion. The results for the 4 to 5 year olds indicate that sex differences in computer use *can* take place with younger children but at this age these differences can be relatively easily modified. It would take a careful study to determine the exact reason for the dramatic difference between the results in September and the results in November-December for this group.

One possible explanation is the attitudes of the teachers working with this group. The 4 to 5 year old group has a different supervising teacher in the afternoon, whose attitudes toward gender roles and computers may be different than the morning teacher's. Another suggested explanation is that the boys dominated the computer early in the year, but as the novelty wore off, the girls felt more comfortable about approaching the computer. One other possible factor was the change of setting from a self-contained room to a larger open space with more children and adults. Finally, by November new software had been acquired that might have been more appealing to the girls.

Outside influences could also have played a part in the initial reluctance of the older girls to choose the computer area. As Kiesler et al. (1983) have noted, "the culture of computing is overwhelmingly male". Much of computer software has been designed by males for males, as shown by the predominance of male figures in programs, computer ads, and on the software packaging. Aggression is often a theme in the programs and in the language of programming (e.g., "kill", "abort", and "boot" are common computer terms). Because of the public stereotype of computers being used by men, many parents may be willing to buy a computer for their son but not for their daughter. It is interesting to note that the one girl who used the computer during the three week period in September has a computer at home. In addition to differential buying practices, when a computer is introduced into the home it is more likely to be used by the father than the mother, thus creating negative role models for daughters.

The long term consequences of sex differences in computer use at the preschool level are profound. They are likely to widen the gap found in the higher grades of the school system, and contribute to continuing male dominance in computer use in the workplace. While many women will have some contact with computers in their work, it will continue to be at the low paying secretarial and clerical jobs, while men will continue to dominate in programming, systems analysis and computer management.

For those who are using computers with young children and want to ensure equity between the sexes in computer use, the following strategies may be helpful:

- *Change the image of the computer centre as "male turf"*. Even in an apparently neutral situation where all the children are asked what they would like to do, the boys can dominate the computer by being asked first because they show more eagerness or because the teacher expects that they will want to use the computer. Strategies boys have used to ensure high computer use include hanging around the computer so that the teacher gets the message, or showing overt hosility to the girls who might be interested in the computer. It might be necessary to ask the girls first, or have "girls' days" and "boys' days" to ensure equal time on the computer (Stalker, 1983).

- *Change the image of the computer itself.* Children need to learn that a computer has many uses. If it is used mainly to play arcade style games, it might not appeal to the girls who have already been socialized into having defined "female" interests. Through pictures, stories and visitors, the children can learn that both men and women use computers in a wide variety of ways at home and at work. Beware of selling the computer to the girls exclusively on the basis of projected female interests or the result might be similar to the grade 7 girl who said that when she grew up she thought she would use the computer to pick a husband and store recipes (Rampy, 1984).

- *Review software for sex bias.* Find programs that are of interest to girls and which promote a positive self-image for females. Beware of sex bias in using apparently neutral programs; for example, using male pronouns to talk about the turtle in LOGO (Sanders, 1984).

- *Provide computer workshops for parents and teachers.* Many teachers and parents are afraid of computer technology. They need a forum to ask questions and be exposed to the issues around computer use. If equity in computer use is raised in these workshops, attitudes that might be influencing children's choices can change.

The computer by itself is non-discriminatory. But the messages the children pick up from software, advertising, teachers and parents can lead to unequal use of computers by boys and girls. Fortunately, it seems that with some effort this can be changed, and equity between the sexes in computer use is possible when working with young children.

REFERENCES

Fisher, G. "Access to Computers", *The Computing Teacher*, 1984, Vol11(8), pp24-27.

Kiesler, S., Sproul, L. and Eccles, J. "Second Class Citizens?", *Psychology Today*, March 1983.

Lockheed, M. and Frakt, S. "Sex Equity: Increasing Girls Use of Computers", *The Computing Teacher*, 1984, Vol11(8), pp16-18.

Miura, I. and Hess, R. "Enrollment Differences in Computer Camps and Summer Classes", *The Computing Teacher*, 1984, Vol11(8), pp22-23.

Piestrup, A. *Preschool Children Use an Apple II to Test Reading Skills Program*, 1981, ERIC ED 202 476.

Rampy, L. *We Teach Children: Computer Literacy as a Feminist Issue*, 1984, ERIC ED 240 028.

Sanders, J. "The Computer: Male, Female or Androgynous?", *The Computing Teacher*. 1984, Vol11(8), pp31-34.

Silvern, et al., 1982, cited by Rampy, L. *We Teach Children: Computer Literacy as a Feminist Issue*, 1984, ERIC ED 240 028.

Sprigle, J. and Schaefer, L. "Age, Gender, and Spatial Knowledge Influences on Preschoolers' Computer Programming Ability", *Early Child Development and Care*, 1984, Vol14, pp243-250.

Stalker, S. *Computers in the Classroom: A Feminist Issue*, 1983, ERIC ED 240 015.

Must Machines Mechanize?: Human Issues In Educational Computing

Dr. William Higginson
Massachusetts Institute of Technology
Arts and Media Technology
E15-312 MIT, Cambridge MA 02139

This paper notes briefly the positions of some thinkers from the humanities and social sciences who have written about the impact of technology. Some implications of these positions for education are suggested.

One of the most striking educational developments of the first half of this decade has been the advent of the various devices of information technology in forms which have made them feasible tools for widespread use in educational settings. The responses of educators to this development have varied greatly. Many teachers have reacted enthusiastically and have expended a great deal of energy in investigating ways of using this technology in their classrooms. The majority of teachers has, however, adopted positions ranging from a cautious, "wait and see" attitude to apprehension verging on phobia. The factors which combine to determine any particular teacher's attitude are almost certainly numerous and diverse. It seems clear, however, at least if one is to make a judgement on the basis of the popular literature, that very few people arrive at their position as the result of any extensive critical analysis of the field.

Recently, largely because of the media's 'discovery' of Artificial Intelligence, there has been some discussion of the human impact of technology. While it would seem not to have had a very significant influence on educational decision making, this topic has a long and distinguished intellectual history. Some of the most active thinkers of this century have made the focal point of their careers a consideration of the ways in which the lives of humans are altered because of the existence of technology. The purpose of this paper is to note (very briefly because of circumstantial constraints) some features of that work and to suggest a few of the implications for educators which emerge from it.

The 'explaining water to fish' characteristic of this task and the problems it presents have been noted by the distinguished Canadian philosopher, George Grant (1969), "We can hold in our minds the enormous benefits of technological society, but we cannot so easily hold the ways it may have deprived us, because technique is ourselves. ... It is difficult to think whether we are deprived of anything essential to our happiness, just because the coming to be of technological society has stripped us above all of the very systems of meaning which disclosed the highest purposes of man..." (p. 137).

The strong spiritual element in Grant's writing is just one of many things he has in common with the French scholar Jacques Ellul. In a number of books, most notably The Technological Society and The Technological System, Ellul has claimed that technology rather than nature has become man's environment. He also argues that it is naive to think of the influence of technology only in terms of its material manifestations. He uses the term "technique" to refer to any standardized means of achieving a predetermined objective. Technological Man, in Ellul's view, is committed to finding, by rational and deliberate means, "the one best way" to solve any problem.

The North American counterpart of Ellul, not least because of the parallel between his term "technics" and Ellul's "technique", is the American polymath Lewis Mumford. For over fifty years through books and essays on architecture, planning, history, philosophy and sociology, Mumford has struggled to show the difference between "the good life and the goods life". While his overall perspective is more secular than either Grant or Ellul, his views on the ultimate emptiness of machine-age life are no less severe. He claims that American society is composed of people who, "eat, drink, marry, bear children and go to their grave in a state that is at best hilarious anesthesia, and at its worst is anxiety, fear, and envy, for lack of the necessary means to achieve the fashionable minimum of sensation".

One contemporary social critic whose views did come to the attention of educators was Ivan Illich. This was because of the savage condemnation of formal systems of instruction contained in his book Deschooling Society. What was not fully appreciated in most educational circles was that Illich saw schools as only one example of the problems which accrue to industrial societies because of the priorities they accord to people and to capital. Writing with a particularly acute sensitivity to the difficulties which arise when these ideas are exported to developing countries, he was perhaps even more fierce in his critique of medical institutions.

The attacks of Mumford, Ellul and Illich have largely been directed against classical forms of technology. Their remarks and those of many others, for instance, the historian Daniel Boorstin who has claimed that the threefold effect of technology is to homogenize, centralize and isolate, have resulted from reflection on artifacts such as looms, lathes, electric motors, cars and telephones. By and large they have had little to say about contemporary technological developments. (It must be said here that relative to the obvious and immediate hazards inherent in biotechnology and nuclear technology, both information technology and classical technology look positively benign.) The Mumfordian tradition has been carried on in the area of information technology by the MIT computer scientist, Joseph Weizenbaum, whose Computer Power and Human Reason has become one of the most widely -cited books of its type. Central to his argument in this work are his attacks on "instrumental reason" and on computer "hackers" who Weizenbaum sees as being caught up in "megalomaniac fantasies".

What influence should the ideas of thinkers like the ones mentioned above have on educators wrestling with the question of how best to use tools like computers in the educational process? One implication is that these ideas need to be taken seriously because critically important issues are at stake. These tools are undoubtedly the most powerful ones to have surfaced since the advent of mass education but it is not at all obvious how they can best be utilized. The insights of these thinkers concerning the long term impact of previous forms of technology might enlighten us in this instance. Having said initially that educators have, by and large, not considered these ideas in much depth, it is worth pointing to one of the exceptions to that general statement if only to illustrate the difficulties inherent in the task. The Summer 1984 issue of Teachers College Record was a special number dedicated to "The Computer in Education in Critical Perspective". The eight articles in that number ranged in quality from competent to irresponsible. All of them seemed totally out of touch with both children and the state of contemporary schooling, several were narrow and unimaginative and one, "The Technological Threat to Education" by Robert J. Sardello, would seem to have lettle place in a serious scholarly journal. For example, "the destructive power of the computer is to be found in the fact that it transforms education into psychology ... we will be transformed into a culture of psychopaths" (p. 631).

A second implication is that if we are concerned about the dangers of dehumanized education, it behooves us to identify more fully the characteristics of humanized education and to see that they are implemented in educational settings whether or not the presence of some technological device is in question. There is considerable evidence from research studies to show that computers can lend themselves extremely well to the support of the social, symbol-creating and tool-using abilities of our species.

In short, we must strive to make computers "convivial" educational tools. This term is used by Illich (1973) to refer to tools which "give each person who uses them the greatest opportunity to enrich the environment with the fruits of his or her vision ... tools foster conviviality to the extent to which they can be easily used, by anybody, as often or as seldom as desired for the accomplishment of a purpose chosen by the user" (pp. 22 -23). The challenge to concerned educators has been articulated by Ellul (1981), "We cannot bring up our children as though they were ignorant of technology ... we must teach them to live in technology and at the same time against technology. We must teach them whatever is necessary to live in this world and, at the same time, to develop a critical awareness of the modern world." (p. 83).

References

Ellul, Jacques, Perspectives on Our Age:Jacques Ellul Speaks on His Life and Work. Edited by William H. Vanderburg. Toronto: Canadian Broadcasting Corporation, 1981. 0887940951.

Grant, George, Technology and Empire: Perspectives on North America. Toronto: House of Anasi, 1969. 088784605X.

Illich, Ivan, Tools for Conviviality. New York: Harper and Row, 1973. 0060803088.

Sloan, Douglas, Ed. Teachers College Record. Volume 85, Number 4, Summer 1984, "The Computer in Education in Critical Perspective". 01614681.

Educational Programming in 'c': A Toolkit Approach

Robert S. McLean
Department of Measurement, Evaluation, and Computer Applications
The Ontario Institute for Studies in Education
252 Bloor St. West
Toronto, Canada M5S 1V6

Productivity of any programmer or designer can be improved dramatically by making use of previous creations -- theirs or others'. Accumulated useful techniques and program code can be formalized into "tools" for programming and design. Such tools can be packaged into "toolkits". The Ontario Approved Educational Microcomputer has stimulated the development of several such "toolkits"; this paper describes the approach, with a specific example of the use of the CEMCORP toolkit for the ICON.

A *tool* is a device which can be used to accomplish a task. The tools and tasks we will be considering are those related to programming of an educational microcomputer. The examples we will use will be from the Ontario Approved Educational Microcomputer (OAEM), and specifically the CEMCORP ICON.

THE IDEA OF A TOOL

The idea of "tools" is a fuzzy one. Our understanding of "tools" may be helped by reference to more physical tools one might find in a handyperson's shop. *Tools* have a number of attributes:

- *Wide applicability:* Tools are usually applicable to a wide range of specific tasks. Tools may be physical artifacts, such as hammers, saws, and screwdrivers. Clearly tools may also be abstract; the handyperson may employ arithmetic and measurement conventions and speak of them as "part of the tools of the trade".
- *Assistance:* Usually (though not universally) tools make the job easier, quicker and more professional. Tools give us "leverage" or multiplied power to accomplish the task.
- *Accumulated knowhow:* Tools represent the packaging of invention and development. Tools have been refined and diversified over time, and it is the accumulated evolution of the tool which gives its potential power in our hands. Consider the hammer: the caveperson probably employed any rock which was handy. Today, our tasks are more refined, and much more suitable variations on the strike-it-with-a-rock theme are available.
- *Cost:* Often, the advantage of using a tool has a price attached: The price may be literal -- it may cost money to obtain the tool; one is paying for the accumulated development as well as for the raw material and its transformation into the artifact. The "price" also includes the additional effort or skill development necessary to capitalize on the power which the tool offers its user.

In the computer domain, the word processing package is a prime example of a tool. It illustrates our points:

- *Applicability:* Word processors are applicable to many specific tasks -- prose, verse, tables of numbers, memos, etc.
- *Assistance:* Writing is often accomplished more quickly and professionally with a word processor, particularly when there is a succession of revisions. But pencil and paper may be more appropriate for short, informal memos.
- *Accumulated knowhow:* A word processor represents the accumulated

knowledge of conventions about the presentation of text, as well as the author's programming and design expertise.

- *Cost:* The "price" includes not only the cost of production and development; there is a "cost" to the user who must learn its conventions and applicability.

In this paper, we consider programming and design activities related to educational computing. The ideas are applicable whether one refers to educational materials (lessonware), utility programs, or even student programming.

The essential idea is that portions of the tasks performed by computer programs can be separated from the specifics of that program, and generalized and codified in a way that makes them useful in the development of other programs.

We can consider two types of software "tools". Sometimes we are interested in the mechanism for the creation of the program, and perhaps this comes closest to the traditional idea of tool. Examples of this type of tool would include text editors, language compilers, problem generators, and similar tools. This category of tools is relatively well known and compartmentalized. Some relatively simple programs -- such as programs to count words in text or to make an index -- often are useful tools. A further exploration of this notion is given by Kernighan and Plauger (1981). Their examples are particularly powerful when a mechanism exists for interconnecting simple programs to perform more elaborate tasks, as it does in UNIX and in QNX on the ICON.

Sometimes we are concerned with what appear to be raw materials or partially finished parts. These tools are subroutines and functions which are cast in a general form, suitable for tailoring to the particular program application we are developing.

It is this latter form of tools which occupies the rest of the paper.

COMBINATION OF TOOLS INTO TOOLKITS

Accomplished software developers know that division of the program's activity into small, meaningful modules is an effective approach to program implementation. If some of those modules can be "off the shelf", productivity is increased manyfold. Each module which is already programmed and tested represents a combination of previous expertise, experience, and advantage similar to what we described for tools.

Productivity of both programmer and designer can be increased if such off the shelf modules exist. The programmer doesn't have to re-invent (and worse, test) those modules; the designer has an extended repertoire of proven programming options from which to choose.

Both CEMCORP and the Ontario Ministry of Education have created collections of routines to help the software developer. These collections are called "toolkits". The specific aim of these is to increase the software developer's productivity, but the ideas are equally applicable to other situations and users. For example, computer studies students can make their programs more impressive and powerful without getting bogged down in detail. They would have the experience of doing more powerful (and useful) projects, addressing higher level design issues.

The components of a useful toolkit, then, are pieces of program which can be used in a large number of other programs, which do useful tasks, and which can be easily tailored to the end user's program.

Toolkit development is a challenging task. While it is difficult to write a program to do intricate things, such as displaying icons on the screen or recognizing what item on the screen has been selected by the program's user, it is much more difficult to reduce the computer's task to a generic function, powerful enough to be useful in many other programs, yet simple enough to be comprehensible to a range of programmers and designers. This is the goal of toolkit designers, however.

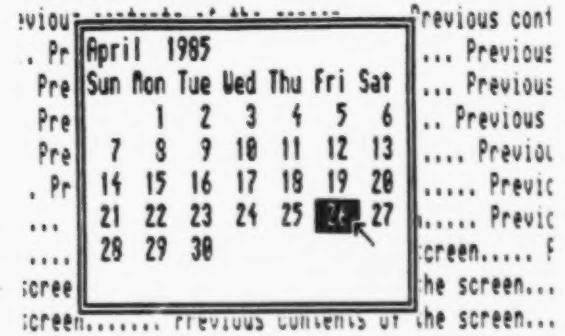
AN EXAMPLE: THE CEMCORP ICON TOOLKIT

Both the Ontario Ministry of Education and CEMCORP have developed toolkits, principally to facilitate the interaction between the user and the program. Our example will illustrate part of CEMCORP's "tool_kit" software (MacKay, *et al.*, 1984), which includes 'c' language functions in the following areas:

- *Window management functions* allow the subdivision of the screen into smaller areas ("windows") for the presentation of text or graphics.
- *Tracker support functions* manage the "pointing" activities directed by the trackball and help the user's program determine what item is selected by the user, through sensitive areas of the screen known as "pick areas". These functions also facilitate the presentation and manipulation of little images (icons) which are often used in that selection.
- *Data management functions* simplify the manipulation of textual data within the program, by providing generic data storage and editing capabilities.
- *Speech generation functions* provide an interface to the speech synthesis capability of the ICON.
- *User input functions* provide pop-up menus, a mechanism for selection between alternatives. A one-line text editor also offers a powerful means of input of responses such as file names or short messages.

AN EXAMPLE USE OF THE TOOLKIT

Here is an example use of CEMCORP's toolkit to draw a month's calendar at an arbitrary location on the computer's screen and allow a user to point to a date, as illustrated here:



April 1985

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Such a function might be useful within an instructional program, an administrative utility, or a computer studies assignment. This function (`draw_calendar`) is another tool that could be included in a toolkit.

The function begins with a global declaration of the month names:

```
char months[] = "January February March April May June
July August September October November December";
```

Next, the function is declared, with its parameters and a description of their types:

```
int draw_calendar (cur_wid, x, y, month, year)
  int cur_wid, x, y, month, year;
```

Next we declare the variables which we will use in this program. These include a number of integer (`int`) variables, a pointer to a string of characters (`char *`), and a file pointer (`FILE *`) which allows us to use a window for output, as if it were a file:

```
{
  int days, i, wid, day_loc_x, day_loc_y;
  char *cp;
  FILE *cal_ptr;
```

The first call to the window manager creates a window in which our calendar will be shown:

```
 wid = create_window (cur_wid, x, y, 28 * CHAR_WIDTH,
    8 * CHAR_HEIGHT, "bfbfb");
  if (wid <= 0) return (0); /* can't create window; return 0 */
```

The window's position is determined by the parameters x and y, which are provided by the applications program. Its size is given by the next two parameters, which are relative to the character size. The final string "bfbfb" denotes the type of outline to place around the window, where b denotes a frame of background color pixels, and f denotes foreground pixel lines. `create_window` returns the identifier for the window which has been created, which we store in the variable `wid`.

The contents of a window can be positioned on an arbitrary coordinate scheme called "user coordinates". In this case we specify a window which is 7 units wide (for the days of the week) and 8 units high (corresponding to the month, day, and six rows of the calendar):

```
 set_user_coord (wid, 0, 0, 7, 8);
  set_window_coord_type (wid, USER);
```

The next several statements are peculiar to the application of drawing a calendar. We determine the coordinate location of the first day of the month and do an `fopen_window` so that we can write textual material to the window as if it were a file or the entire screen. We then use normal output commands such as `putc` (put

BEST COPY AVAILABLE

out a character) or `fprintf` (formatted printing) to write the headings of the calendar.

```
day_loc_y = 5;
  day_loc_x = day_of_month(year, month, 1);
  cal_ptr = fopen_window(wid, "w");
  putc(CURSOR_OFF, cal_ptr); /* remove cursor */
  putc(CS, cal_ptr); /* clear screen (in window) */
  days = 31; /* decide how many days in this month */
  if(month = = 4 || month = = 6 || month = = 9 ||
    month = = 11) days = 30;
  if(month = = 2) days = (year % 4 = = 0 &&
    year % 100 != 0 || year % 400 = = 0) ? 29 : 28;
  cp = months; /* print month heading */
  while(month > 0) if(*cp + +  = = ' ') month--;
  while(*cp != ' ' && *cp != '\0') putc(*cp + +, cal_ptr);
  fprintf(cal_ptr,
   " %d\nSun Mon Tue Wed Thu Fri Sat\n", year);
```

Each day is output in succession by a for-loop. This loop moves the printing cursor to the position of the day's number and prints it. More interestingly for our present discussion, it also establishes a "pick area" in the area corresponding to that day:

```
for(i = 1; i < = days; i + +)
   {
   position_text(cal_ptr, 4*day_loc_x + 1, day_loc_y);
   fprintf(cal_ptr, "%2d", i);
   create_pick_area(wid, day_loc_x, day_loc_y, 1, 1, i,
      HIGHLIGHT_WHEN_ENTERED);
   if( + +day_loc_x > 6) day_loc_y--;
   day_loc_x = day_loc_x % 7;
   }
```

The function `create_pick_area` is found in the tracker support portion of the Toolkit. It allows one to specify rectangular areas of the screen which will be sensitive to the user's pointing by use of the trackball to move a 'tracker" around the screen. In our case, a pick area for each day is located at that day's number. It is one unit wide and high (in the arbitrary user coordinates corresponding to one day on the calendar), and will be highlighted automatically when the tracker enters that area, giving visual feedback of the potentially selected area. The pick area is assigned a number (given by the for-variable i) which can be used later to find out which day has been selected. The remainder of the for-loop simply adjusts the positioning to print the numbers in calendar format.

The remainder of this function simply finishes up the drawing of the calendar by flushing any remaining characters out to the screen and disconnecting the window as a text output device with `fclose_window`. Finally the function exits by returning the identification of the window holding the calendar so that other processes can refer to that window.

```
fflush(cal_ptr);
 fclose_window(cal_ptr);
 return(wid);
 }
```

We can illustrate the use of `draw_calendar` in a simple main program as follows:

```
#include "/tool_kit/win_mgr.h"
#include <stdio.h>
#define ACTION1_PRESS 0x8d
#define ACTION2_PRESS 0x80
#define CHAR_HEIGHT (7500/24)
#define CHAR_WIDTH (10000/80)
  int cal_window;
main(argc, argv) /* arguments are x y month year (x and y
    in 1/10000 ths), for example, cal 1000 1000 4 85 shows
    a calendar for April 1985 with its lower left corner
    located at coordinates 1000,1000 */
  int argc; char *argv[];
  {
  int saved_window_id;
  set_option(stdin, 0x1000); /* keyboard sends all characters */
```

The first several lines are housekeeping which serve to set up some of the context for this program. The `main` function gets its arguments from the command line. These will be the x and y position (in one ten-thousandths of a screen width), the month and the year. The initial portion of the function defines those arguments and internal variables. The `set_option` tells the keyboard processor to pass along all characters directly to the program without performing its usual editing functions.

```
 start_win_mgr(); /* start window manager task running */
  saved_window_id = save_window(0); /* save screen */
```

We start the window manager because the support for windows is a separate task; this must be done before any window manager routine is called. To be more dramatic, we use another tool (`save_window`) to save the entire screen so we can put it back after this program has run thereby not disrupting whatever is already on the screen.

Next we use the function we created, to draw the calendar for the month and year specified, at the given coordinates:

```
 cal_window = draw_calendar(0, atoi(argv[1]), atoi(argv[2]),
    atoi(argv[3]), atoi(argv[4])); /* draw the calendar */
```

At this point the calendar has been displayed on the screen. Our job is done, but we might as well have some additional fun with it to illustrate some other tools. We will now allow the user to select a date from the calendar and, by pressing the ACTION key, pop-up a menu which (in the real applications program) would lead to the actual function to be performed by

BEST COPY AVAILABLE

the program.

First we move the tracker (the arrow whose movement is controlled by the trackball) into a suitable area at the upper right-hand corner of the calendar and display it:

```
move_tracker_to(cal_window,
   sample_window_xs(cal_window, cal_window, NORM),
   sample_window_ys(cal_window, cal_window, NORM));
 constrain_tracker(cal_window);/*keep trkr in window*/
 display_tracker();
```

Now we are ready to process the user's inquiries, using a separate function:

```
process_inquiries(); /* do what the user wants to do */
```

Having finished processing the user's commands, we continue in our `main` function, to clean up after ourselves:

```
delete_window(cal_window); /* clean up after program */
 erase_tracker();
 restore_saved_item(saved_window_id);
 delete_saved_item(saved_window_id);
 stop_win_mgr();
 }
```

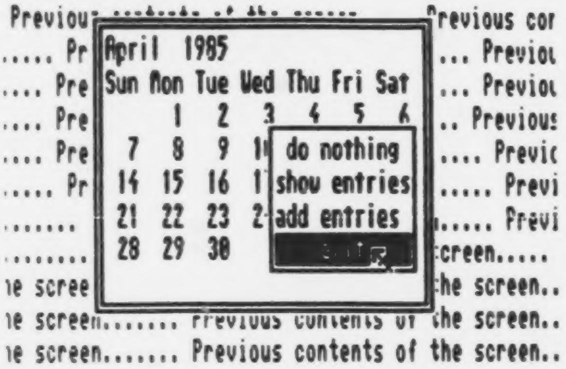
We delete the calendar window, erase the tracker, restore the entire screen, thereby restoring the display to the way it was, and delete the saved window from the storage automatically maintained for us by the window manager. Finally, because this is the end of the program, we disconnect the window manager task and recover all of the space used by these facilities.

The function `process_inquiries` forms the basis for user selection of the computer's function in the calendar. It illustrates one of the user input tools, `pop_menu`:

```
process_inquiries()
  {
  int choice;
  do{
    wait_for_action_key();
    if(sample_tracker_window(0) = = cal_window)
      choice = pop_menu(cal_window, ".do nothing.show
              entries.add entries.exit");
    }while(choice != 3);
  }
```

This function waits for the user to press the ACTION key and then pops-up a menu having four choices on it. In a real application, this function would be expanded to perform the choice which has been selected. In the example, the only choice currently programmed is the one to exit (choice = 3), at which point `process_inquiry` exits and we return to the `main` function.

The popup menu appears on the calendar window as shown here:



```
Previou                                  Previous co
..... Pr April  1985                        ... Previou
.... Pre Sun Mon Tue Wed Thu Fri Sat        ... Previou
.... Pre       1   2   3   4   5   6        .. Previous
.... Pre   7   8   9  1 do nothing          .... Previc
..... Pr  14  15  16  1 show entries        ..... Previ
.......   21  22  23  2 add entries         ..... Previ
........  28  29  30                        creen.....
ie scree                                    the screen..
ie screen....... Previous contents of the screen..
ie screen....... Previous contents of the screen..
```

THE TOOLKIT CHALLENGE

Toolkits allow software designers and programmers to work at a high level. The existing toolkits are a tremendous help with the user interface. The toolkit challenge -- in an educational computing environment -- is to extend these powerful ideas to other areas of educational software creation.

REFERENCES

Kernighan, B.W. and P.J. Plauger (1981) *Software Tools in Pascal.* Reading, Mass.: Addison-Wesley.

MacKay, R., J. Likuski, S. Hind-Smith, P. Vytas, and H. Nienhuis (1984) *Tool Kit for the ICON.* Toronto: CEMCORP.

OTHER ROUTINES USED IN EXAMPLE

```
wait_for_action_key()
  {
  unsigned c;
  do c = getc(stdin);
    while(c != ACTION1_PRESS && c !=
ACTION2_PRESS);
  }
int day_of_month(year, month, day)
  int year, month, day;
  { /* CAUTION!! this function not fully tested*/
  int c, m, d; /* based on Zeller's congruence */
  m = month - 2 ;
  if(month < 3){ m + = 12; year--; }
  d = year % 100;
  c = year / 100;
  m = (26 * m - 2 )/ 10;
  return(( m + day + d + d/4 + c/4 - 2*c) % 7);
  }
position_text(file_ptr, x_loc, y_loc)
  FILE *file_ptr;
  int x_loc, y_loc;
  { /* output TELIDON codes to position text cursor */
  putc(APS, file_ptr);
  putc(y_loc + 32, file_ptr);
  putc(x_loc + 32, file_ptr);
  fflush(file_ptr);
  }
```

BEST COPY AVAILABLE

Review of "Learning and Telecommunications in New York State: Action Recommendations"

Peter Stoll, Ph.D.
New York State Education Department
Center for Learning Technologies
Room 9A47 - Cultural Education Center
Albany, New York 12230

During 1983-84, the New York State Education Department's Center for Learning Technologies has developed fifteen "action recommendations" for implementing the learning and telecommunications technologies within schools, libraries, museums, colleges, and vocational rehabilitation institutions. These recommendations address Regents priority goals of excellence, productivity, and equity. The recommendations were derived from learning technology issues, including research and development, professional development, quality instructional material, telecommunications, and technological integration.

Modern society is undergoing profound technological and social changes brought about by "the information revolution." The impact of this revolution affects individuals, institutions, and nations. If people are to thrive in a world shaped by these developments, they must adapt through education. The challenge facing local, state and federal educational authorities is the need to define new types of education and training strategies to enable institutions to respond to these changes.

In New York State, this activity began when the Board of Regents established the Center for Learning Technologies in 1982, a central administration unit situated within the staff of the Office of the Executive Deputy Commissioner of Education. The Regents charged the Center with the responsibility of creating a statewide plan for implementing learning technologies across schools, libraries, museums, vocational rehabilitation institutions, public television and radio network, and postsecondary institutions.

During 1983-84, Center staff developed a number of proposed "action recommendations" to serve as a framework for formulating specific agenda and operational plans to implement the technologies in those sectors of education for which the Department has responsibility - elementary and secondary education, cultural education, vocational rehabilitation, and higher education. These actions are currently being reviewed by the Department's Executive Staff, the intra-Departmental Learning Technologies Task Force, and the Regents Advisory Council for Learning Technologies. The timeline for completing the plan calls for final agreement on the action recommendations by March 1985 and the development of specified timeframes and plans for Departmental Offices during 1985-1986.

The learning technology "action recommendations" proposed for New York State's educational and cultural system address three central Regents priority goals for education.

Excellence

Interactive learning technologies must be used to achieve quality within our educational system and these technologies must reflect the highest standards.

Productivity

Interactive learning technologies must be employed to achieve greater efficiency in the use of educational resources and enhanced professional productivity.

Equity

Technological resources, including the information conveyed by these technologies, must be equitably distributed to all individuals, groups, institutions, and regions.

The action recommendations, which address the above goals, have been specifically derived from a study of key technology issues relating to research and development, staff development, telecommunications, integration, and quality materials.

They were derived from similar issues discussed in the Center's initial concept paper, "Technology and Education," which was presented to the Board of Regents in June, 1983.

During the preceding year, the Regents' Advisory Council on Learning Technologies held extensive discussions on these issues and has recently proposed significant recommendations pertaining to telecommunications and staff development to the Commissioner of Education.

They were identified as priority issues through a series of fact-finding discussions Center staff held with representatives of the major Education Department program offices.

They have been consistently identified as national priority technology issues in a series of national and regional meetings involving staff of the Center for Learning Technologies, the Council of Chief State School Officers, and designated State Education Department technology leaders.

They were extensively documented and addressed as the major topics of discussion at the First National Educational Technology Leadership Conference.

The issues are as follows:

Research and Development

Decisions on implementing technology within programs and services must be predicated upon a solid base of information. Research and development activities must address the application of the technologies to meet the needs of education, the importance of evolving a variety of means of disseminating information about technology to field constituencies, and the need to conduct systematic evaluations of the interactive technologies to insure appropriateness and effectiveness.

Professional Development

The most important guarantee that the learning technologies will be appropriately integrated within schools, museums, libraries, colleges and universities, adult educational centers, and vocational rehabilitation sites is the adequate provision of effective preservice and inservice training to students, professionals, and other staff. This issue addresses the need to implement effective staff development programs on both the instructional and administrative uses of the learning and telecommunications technologies. Professional development must also include a planning component for the most effective use of learning technologies.

Quality Instructional Materials

There is growing evidence that the current quality and availability of higher-order problem-solving learning technology materials is inadequate. This issue concerns the need to enhance the quality of learning technology resources and to insure that there is a close match between curriculum priorities of the State Education Department and the content of computer or other interactive programs.

Telecommunications

Electronic networks have the potential to achieve the goals of equitable and enhanced delivery of educational and cultural resources with increased efficiency and cost-effectiveness. However, the rapid evolution of telecommunications systems, the continual deregulation of rates and laws, and the existence of "untapped" capacities, require that greater attention be given to planning a comprehensive telecommunications system to support the educational and cultural enterprise.

Technological Integration

It is necessary to integrate technologies within the content and program areas of the educational and cultural institutions to insure that they are effectively used in enhancing the quality of programs. At the same time, new developments in the industry require that the technologies themselves should be integrated to effectively capitalize on their collective strength.

Some examples of the "action recommendations" proposed to address the goals of excellence, productivity, and equity are as follows:

Excellence

Goal: To employ learning and telecommunications technologies in support of enhanced standards and quality for New York's cultural and educational institutions.

Recommendation

The Department will cooperatively develop a learning technologies and telecommunications research and development agenda, which will serve as a basis for seeking State, Federal, foundation and commercial support for technologies initiatives to enhance the quality of our educational/cultural institutions.

Recommendation

The Department will develop a systematic plan and agenda of priority "producer/consumer" forums to provide for ongoing participation of Department and field staff in the development of learning technology materials and to insure a "goodness of fit" between these programs and the technical content concerns of educators and professionals at the local, regional and State levels.

Recommendation

Model learning technology staff development programs will be developed by the Department and delivered in targeted program areas to train a variety of educational and cultural professionals to insure that the technologies are employed to enhance the standards and qualities of educational and cultural programs. These models will address the minimal competencies and knowledge required for professionals to integrate the full range of appropriate learning technologies within their respective disciplines and program areas. Appropriate Department offices will review and modify current and proposed certification programs to insure that minimal competencies and knowledge about the uses of learning technologies are included in the development of model staff development programs.

Productivity

Goal: To insure efficiency in the use of resources and greater productivity in general within our educational and cultural systems through the implementation of appropriate learning and telecommunications technologies.

Recommendation

The Department will expand its capacity for identifying and transferring promising technological developments and applications from the military, industrial, and non-profit sectors that have implications for enhancing the productiveness and efficiency of New York's educational and cultural institutions. This will be accomplished via the following:

o The transfer of emerging technologies and developments from non-educational and cultural sectors to educational and cultural sectors, and;

o The identification of exemplary programs and activities that currently exist within the field and may be replicated for transfer to other educational/cultural institutions.

Recommendation

The Department will continue to work with commercial industries on the development, testing, and implementation of integrated, multifunctional systems (i.e., those systems which combine several different learning and telecommunications technologies and serve a variety of purposes) to insure that a base of knowledge and model (s) exist for the most efficient and effective delivery of service.

Recommendation

The Department will continue to pursue a close relationship with State-level telecommunications efforts, such as the Urban Development Corporation and the Executive Office, to insure high level, comprehensive state telecommunications study and planning. A comprehensive study should address current and planned telecommunications capacities in the implementation of a telecommunications "highway" which might be used by all state agencies in a cost-sharing arrangement.

Equity

Goal: To promote the equitable distribution of technological resources to all populations; to safeguard access to the positive benefits of technology for all; and, to work cooperatively with the producing sector to insure instructional technology materials are non-biased and appropriate.

Recommendation

Arrangements for statewide volume price discounts for quality learning technology hardware and software will be undertaken by the Department to insure that there is greater access for all of New York's citizens to these technologies.

Recommendation

The Department will work with hardware and software producers to address the development and introduction of computer peripherals and software that will enable the physically handicapped, the learning disabled, and the developmentally delayed learners of New York State to access and effectively use learning technologies.

A full discussion of the three goals and the full list of fifteen "action recommendations" is presented in the paper, "Learning Technologies and Telecommunications in New York State: Action Recommendations."

Instructions On How To Prepare Papers For Conference Proceedings

Jo Ann Wilton
Peel Board of Education
Mississauga, Ontario

The conference budget does not allow for type setting and editing every paper for the Proceedings. However, we would like to have a product of which everyone will be proud. The guidelines outlined in this article will help ensure that there is as much continuity from page to page as we can achieve. Please read all of them - at least some will apply to you.

The purpose of this article is to outline the format for papers which are to be included in the Conference Proceedings. Some of you have prepared articles for this form of publication before; but, a significant number will not have. For that reason these suggestions are divided into a set of requirements for papers in this particular document and a set of recommendations for the presentation of articles which may be of help to those of you who have not had a lot of opportunity to have your work printed before.

The following are **requirements** for the style and form of the presentation of your paper.

The paper must be submitted on white bond or white computer paper cut to size in a *fully completed and edited* form. The way you send it in is the way it will go into the Proceedings - spelling misteaks and all!

The paper must not exceed *four pages in length* including any references.

The dimensions of the text presentation are firm - do not go beyond the lines shown on the accompanying *format pages*. (There are two format pages - one with the heading and abstract for the first page of your document and the second for all other pages.)

The main text should be in a *two column format* as is this example paper unless there is a good and logical reason for straying from this mode. The abstract, however, will extend across the full 7 inch space.

The *title and author(s) name(s)* will be added by the editor before printing and will be in the type face used in this example. You need do nothing except be sure that you include an accurate title and appropriate credits on the accompanying form for that purpose.

Because the title is to be added by us, be sure to *start the abstract* at the line indicated for the text on the first page of the article. The other pages should be started at the top line. *No page numbers* should be typed on your paper. Instead, lightly pencil them in the lower right hand corner.

Diagrams or figures must be camera ready quality and reduced and inserted into the location you want them. Photographs can be submitted in a larger size because we will have to have them done separately; but, be sure to indicate *clearly* where it is to be placed in the document and leave sufficient space.

This example has been printed with Scribe on a laser printer. All the additional text in the Proceedings (Forward, Table of Contents, Index, etc) will be done in this format. Your paper will look best if you can come as close as possible to this quality - but, do the best you can. If you are typing the product, use a 12 pitch type face and a new ribbon. If you have a computer word processor:

1. Select the type face closest to that used in this paper and stick to it. Changing the type face too often in an academic document may look 'neat' at the time but will look tacky in the final document and embarassing in a few months time.
2. Use the newest ribbon you have. Don't go to all this trouble and then have your paper unable to be read.
3. Avoid dot matrix printers. If you have no other choice, so be it; but the extra effort to go to your college or school board offices to get some help, may increase your satisfaction with what you attach your name to.
4. Check the spacing of words in the narrow columns. Often a bit of readjustment can eliminate ungraceful gaps in some lines.

If this is your first published article, congratulations and let's make it the best show we can. Here are some **recommendations**:

- Check on the number of words that your presentation format will allow on four pages and write to that size. You can not go over, but you could end in three, if you have to.
- The easiest way to do references, if you have any, is to put the name and date in the text (Wilton, 1984). Then just put your references in alphabetic order at the end of the paper.
- Remember that those who read your piece in the months and years to follow were not at the conference. Make your ideas and arguments clear and self-sufficient.
- If you are describing something that you or your institution has done with computers, explain the situation, tell them what you did, outline any results and then project conclusions. If any further materials or information is available - let them know what and how to get it.
- Any tables or diagrams which are needed for the presentation should be included but should be referred to or called upon in the text.
- Make the article as readable as possible. Give it to a colleague for an honest evaluation.
- If you are doing a presentation which will be largely show and tell - things which can't easily be included in the article, spend time working out what are the essential points of the session and how they can be explained to a reading audience. Not all conference sessions can or should be summarized in text form and yours may be one.

If there are any questions about presentation, please contact the editor as soon as possible. Don't leave it until the day before the papers are due. There will be many papers in these Proceedings but each of you should consider yours the most important one. The better each one is the better we all look who have our names anywhere in the book.

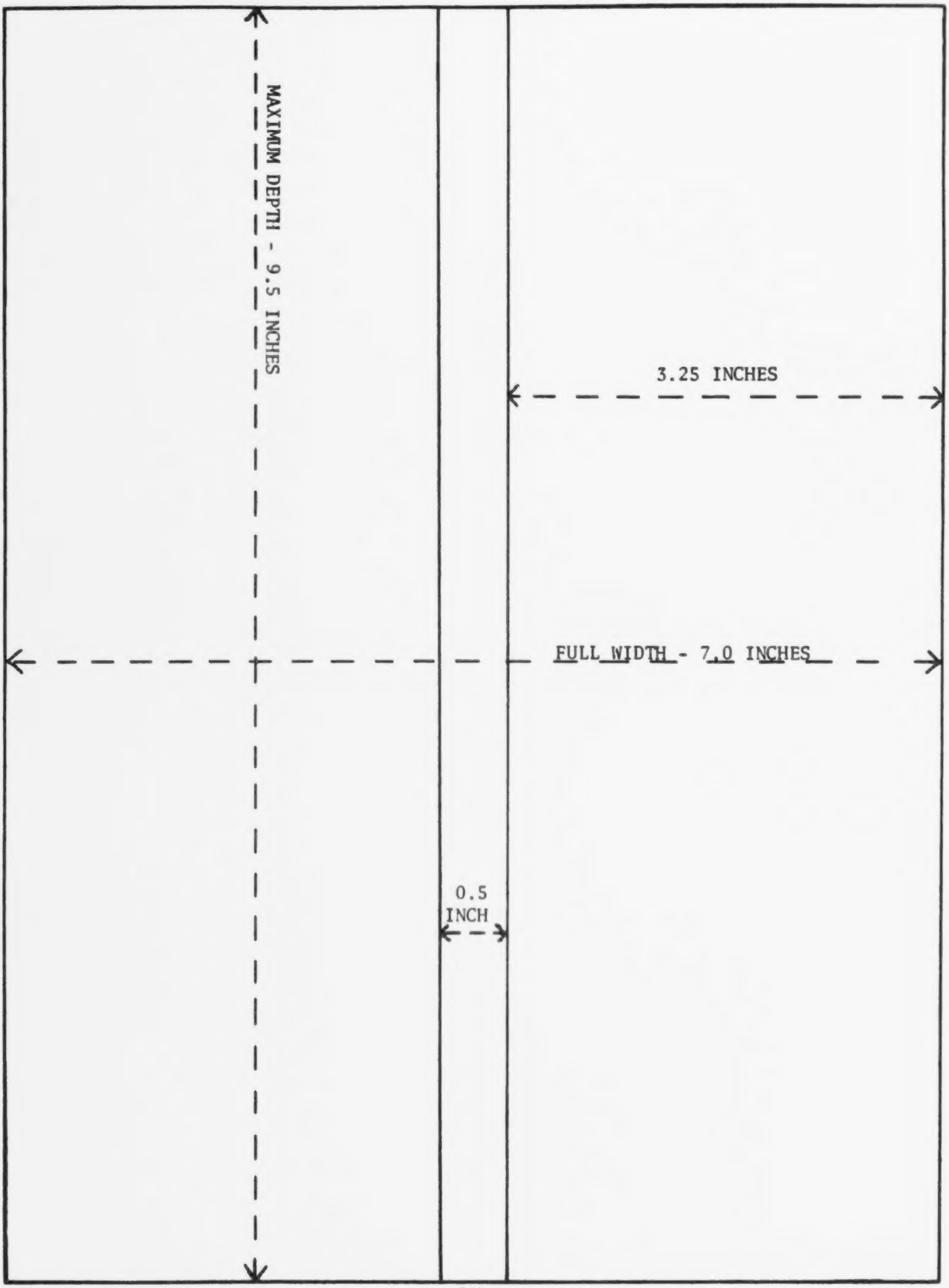
Because the Conference Proceedings are for general sale after the conference by the organizing associations, we require you to sign the release portion of the form provided.

The date for submission is final. All papers must arrive on the date shown *or sooner*. Any articles, which because of poor quality of presentation or content can not be included, will be returned. If time permits, such papers could be resubmitted.

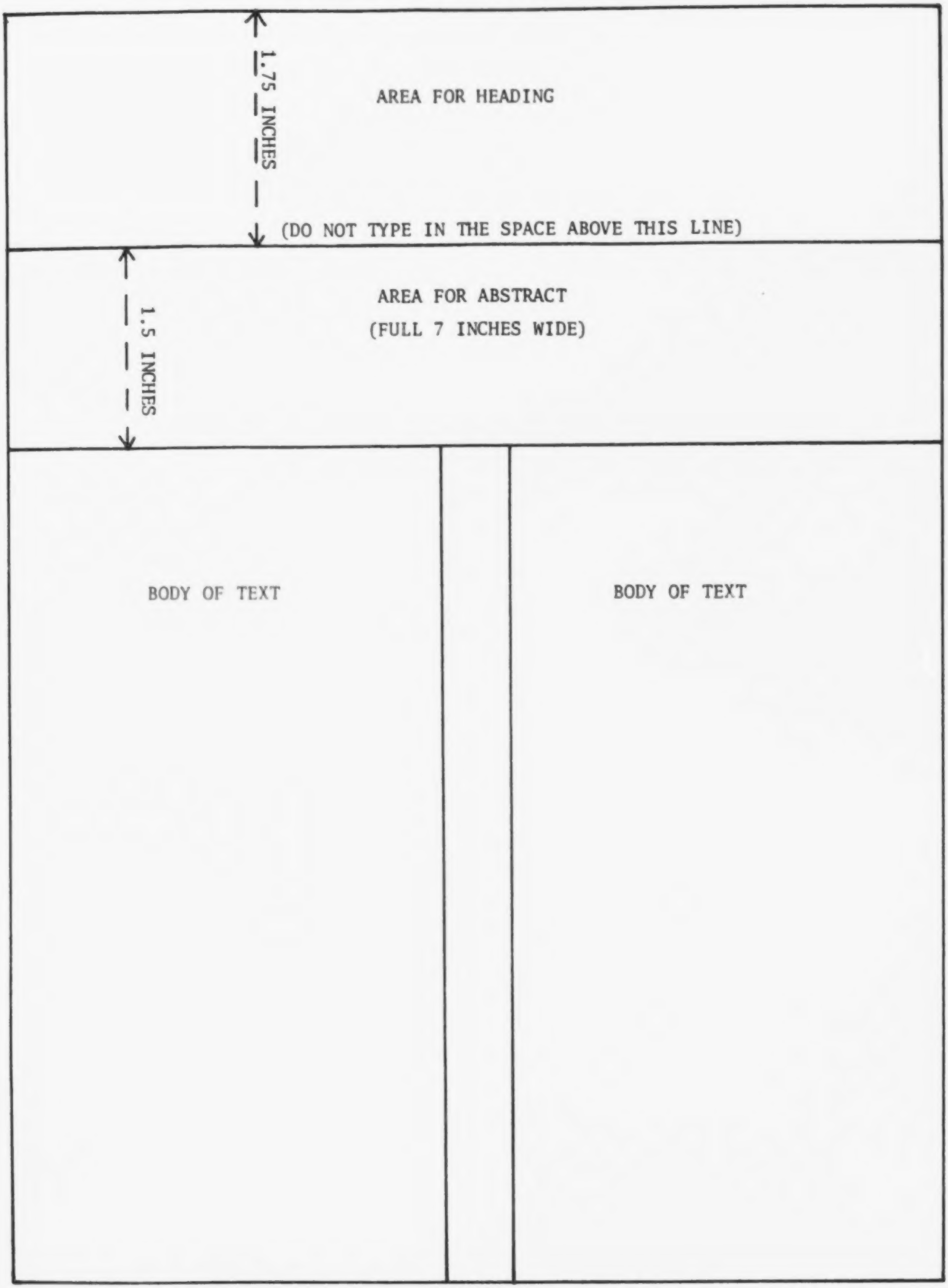
REFERENCES

Wilton, J.A. "A Perfectly Trivial Article", ...Unlikely Journal, 1984, Vol1(1), pp1-2.

FORMAT FOR ALL PAGES EXCEPT FIRST


MAXIMUM DEPTH - 9.5 INCHES
3.25 INCHES
FULL WIDTH - 7.0 INCHES
0.5
INCH

FORMAT FOR FIRST PAGE ONLY


1.75 INCHES
AREA FOR HEADING
(DO NOT TYPE IN THE SPACE ABOVE THIS LINE)
1.5 INCHES
AREA FOR ABSTRACT
(FULL 7 INCHES WIDE)
BODY OF TEXT
BODY OF TEXT

PAPERS BY AUTHOR

ANDERSON, E.L.
Graduate Microcomputer Education: Another Perspective 47

ANREP-MATTHIESEN, F-G.
Computer Applications Across the Business Education Curriculum 275

AVANTI, N.
"Clustering" in Vocational Business Education. 289

BAILEY, M.G.
Acquiring Computer Literacy in Higher Education: A Tennessee Mandate 93

BEDERMAN, F.
Computers and Special Students: Classroom Applications Beyond. 205
Drill and Practice

BENSTEIN, J.H.
Some Thoughts on Data Management for Special Education 220

BHAGALOO, R.K.
Managing Information: Security, Access and Dissemination - A Conceptual
Model and Case Study. 25

BOZEMAN, Wm.C. and OLSEN, C.R.
Expert Systems: Applications in Eduational Research and Decision Making 5

BRINE, J.W. and JOHNSON, E.M.
Educational Computing in Britain. 313

BURGER, M.
Authoring Languages/Systems in K-12 Schools: What's Happening 101

BURNETT, J.D.
It's Almost Time for Logo 147

BURNS, J.
A Model for Inservice Training of Teachers That Effects Real Change 243

BURTON, J.
Integrating Tool Software in the School and Classroom 117

BYRNES, R.
Oracle (Consultant) Science Fair Research Project 130

CIEPLY, S.M.
Kids and Computers - A Summer Program for Elementary Students 67

COHEN, R.
Developmental Studies of Children Learning Logo Programming: the Cognitive Dimension . . . 157

COHEN, R. and BARKER, G.
The OISE Logo Project: Apple/Terrapin Software. . . . 154

CRAWFORD, R.M.
Computerizing the Classroom Teacher's Tasks (Burn Your Computer Out, Not Yourself. . . . 42

DAWLEY, D.L.
A System and Strategy for Data Processing Planning for Universities . . . 21

DIDUR, B.
Computer Studies in Ontario Schools: Curriculum, Hardware and Software . . . 63

DURELL, B.
It May Be Courseware, But Is It Educational? . . . 187

EDWARDS, L.
Planning Computer Labs . . . 38

EVANS, C.L.
Retooling Black Schools of Business for the Information Culture . . . 1

FREEMAN, R.J.
An Assessment of CAI in the Field of ESL . . . 247

FRYE, I. and PARSONS, J.A.
Strategies for Developing Faculty Computer Literacy . . . 277

FUNG, M.
Tips for Teachers on Organizing Data . . . 293

GILLESPIE, F.
An Analysis of Instruction Improvement Projects Utilizing Computer Resources. . . 263

GOATES, W.
Linking Microcomputers to Mainframes for Period Attendance and Local School Flexibility . . . 29

GODDARD, Wm. P.
It Figures: The Computing Component of the Second International Mathematics Study . . . 239

GOFORTH, D.
Robotics with Logo and a Tasman Turtle . . . 161

GOLDSTEIN, G.
Administrator "Homo-Optimus": A Diagnosis of Change and Relationships Affected by Technology 17

GOOD, J.
Elementary Computer Literacy: An Inexpensive Strategy 89

HADDUCK, C.G.
A System for Computer-Managed Instruction 32

Where Would You Go Tomorrow If Disaster Struck Today? 9

HASSELL, J.
Programming Schemes: A Model for Understanding and Teaching Programming, Programming Debugging and Testing 73

HERMANN, B.
Computers and Composition in the Music Education Classroom 109

HIGGINSON, Wm.
Must Machines Mechanize?: Human Issues in Educational Computing 326

HUNTER, J.
An Evaluation of the Course Content of a CIS Curriculum 77

KEPNER, H.S.
Training Teachers of Secondary Computer Science - A Methods Course 59

KING, T. and PARISH, M.
"MacEnglish" - Using the Macintosh for Senior High English and Journalism 283

KOEHLER, T.J., BARR, D., EDMISTER, P. and BOWERS, A.
Individual Educational Plans - A Continuing Saga of Computer Support 201

KREUTZ, D.A.
Getting to the Core of the Apple: Global Computer Development 35

LAND, M. and TURNER, S.
What Are the Effects of Computer Programming on Cognitive Skills? 51

LAPP, S.
Using Low-Resolution Graphics to Teach B.A.S.I.C. 81

LEVINSON, M.S. and WALCOTT, J.A.
On-line Data Bases: A School Project 269

MacDERMOT, H.G.
Minnesota Information Technology Learner Outcomes 279

MALONE, A.P.
Micrometrics: A Metric System Tutorial . . . 194

MARCHIONINI, G.
Spreadsheets and Decision Support Systems: Tools and Environments for Teaching and Learning . . . 127

MARSHALL, J.C. and BANNON, S.H.
Computer Attitudes and Knowledge Among Students and Educators . . . 305

MARTIN, C.D. and HELLER, R.S.
Teacher Training and Logo: Higher Levels of Concern . . . 139

MATTHEWS, J.E.
Diary of a Computer Resource Teacher - Year One . . . 266

McCAHILL, P.
On-line Needs Diagnosis . . . 217

McLEAN, R.S.
Educational Programming in 'c': A Toolkit Approach . . . 330

MEINHARDT, C.
Documentation - A Key to Success . . . 309

METROS, S.E.
Electronic Thinking Cap: Microcomputer Enhanced Creative Problem Solving for Visually Oriented Students . . . 179

MOONT, S.E.
Databases, Bulletin Boards and Information Retrieval: Promises and Pitfalls (Or Should I Buy A Modem?) . . . 133

MORARIU, J.
Human Factors Concerns in Education Software Design, Selection and Evaluation . . . 259

MOURSUND, D.
Problem Solving: A Computer Oriented View . . . 235

NITEFOR, M.
Talking Books - Print Learning Materials Go Interactive . . . 113

NOONAN, L.
Computers and Art . . . 272

OLDS, H.F.
Computing Tools in the Classroom: Through the Curriculum and Beyond . . . 121

OMAN, P.W.
Identifying Student Characteristics Influencing Success in Introductory Computer Science Courses 69

OSBORNE, J.
From Course Requests to Examination Schedule 13

PAINE, J.
Marketing Educational Software: A Publisher's Perspective 175

PARAMSKAS, D.M.
Custom-Designed Lessonware for Second Language Learning 191

PIKE, R.
Looking at Learning from Lessonware 231

RAGSDALE, R.
Research Needs for Computers in Education 297

RIVERS, R. and VOCKELL, E.
A Model for Implementing Computers (K-12) to Promote Learning and Literacy 197

ROBINSON, S.
Introducing Computers into A T.M.R. School: a Proposal 226

ROUSE, B.
Constructing Computer Interfaces for the Laboratory 301

RUBINCAM, I. and SOMER, G.
Characteristics of the Periodical Literature in Computer Applications to Eduation. 318

SPEED, B.
Using Microcomputers to Improve Self Image of Learning Disabled Students 213

SPERO, S.W.
Second Generation Electronic Spreadsheets and the Teacher's Gradebook 105

STOLL, P.
Review of "Learning and Telecommunications in New York State: Action Recommendations" 335

SULLIVAN, G.
Competency Based Computer Education 124

TABOR, F.M.
    An Information Age Plan for Special Educators: Computers and Technology . . . 223

TEMPLETON, B. and GARNER, J.
    The ALICE Programming Education System . . . 85

THOMAS, B.S.
    Alcohol Education: The Choice is Yours . . . 167

    Use of CAI to Improve Instruction in Nursing Research . . . 171

TRUEMAN, D.
    QNX and the ICON - Security that Works . . . 55

TURNBULL, G.R.
    Word Processors or Story Processors? (Redefining Literacy) . . . 209

UPITIS, R.
    Melodic Manipulations: Teaching Composition with Computers . . . 183

WATT, D.H.
    Creating Logo Microworlds . . . 143

WATT, M.
    Teaching Recursion in Logo by Playing Turtle and Playing Procedure . . . 151

WILTON, J.A.
    Local Area Networks: Present and Future Possibilities for Classroom Use . . . 255

    Instructions on How to Prepare Papers for Conference Proceedings . . . 339

WHOLEBEN, B.E.
    Incorporating Computer Simulation Methods Into Classroom Instructions: From Social Studies and Mathematic to Emergency Medicine . . . 251

WOODILL, G., ANDERSON, K. and BERNHARD, J.
    Sexism, Computers and Young Children . . . 323